UML Statechart Fault Tree Generation By Model Checking

L. T. Herbert
KPMG, Copenhagen, Denmark

Z.N.L. Hansen
The Technical University of Denmark, Lyngby, Denmark

ABSTRACT

Creating fault tolerant and efficient process workflows poses a significant challenge. Individual faults, defined as an abnormal conditions or defects in a component, equipment, or sub-process, must be handled so that the system may continue to operate, and are typically addressed by implementing various domain specific safeguards. In complex systems, individual faults may combine to give rise to system failure, defined as a state or condition of not meeting a desirable or intended objective. The safety analysis of such systems is labour-intensive and requires a key creative step where safety engineers imagine what undesirable events can occur under which conditions.

Fault Tree Analysis (FTA) attempts to analyse the failure of systems by composing logic diagrams of separate individual faults to determine the probability of larger compound faults occurring. FTA is a commonly used method to derive and analyse potential failures and their impact on overall system reliability and safety. FTA has seen extensive refinement and widespread adoption and is today considered a proven and accepted reliability engineering technique, often required for regulatory approval of systems. However, fault trees are typically manually constructed and determining the probabilities of faults occurring in systems which exhibit stochastic behaviour in the course of their correct execution is difficult, time-consuming and error prone.

Typically a FTA is based on an informal description of the underlying system, or requires modelling the system in an FTA specific language. This makes it difficult to check the consistency of the analysis, because it is possible that causes are noted in the tree which do not lead to the failure (incorrectness) or that some causes of failure are overlooked (incompleteness).

To avoid these deficiencies, our approach derives the fault tree directly from the formal system model, under the assumption that any state can fail.

We present a framework for the automated generation of fault trees from models of real-world process workflows, expressed in a formalised subset of the popular Business Process Modelling and Notation (BPMN) language. To capture uncertainty and unreliability in workflows, we extend this formalism with probabilistic non-deterministic branching.We present an algorithm that allows for exhaustive generation of possible error states that could arise in execution of the model, where the generated error states allow for both fail-stop behaviour and continued system execution.

herbert

REFERENCES

Crockford, N., 1986, An Introduction to Risk Management, Woodhead-Faulkner, Cambridge England.

Ericson, Clifton A., 2005, Fault Tree Analysis, Hazard Analysis Techniques for System Safety, 183–221, John Wiley & Sons, Inc., New Jersey,USA.

Banach, R. and Bozzano, M., 2011, The mechanical generation of fault trees for reactive systems via retrenchment II: clocked and feedback circuits, Formal Aspects of Computing, 1–49.

Liggesmeyer, P. and Rothfelder, M., 1998, Improving system reliability with automatic fault tree generation, , Fault-Tolerant Computing, 1998. Digest of Papers. Twenty-Eighth Annual International Symposium on, jun, 90–99