Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 29-30, 2009

An Integrated Approach of Model checking and Temporal Fault Tree for System Safety Analysis

Kwang Yong Koh* and Poong Hyun Seong
Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology, 371-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
**Corresponding author: goeric1@kaist.ac.kr*

1. Introduction

Digitalization of instruments and control systems in nuclear power plants offers the potential to improve plant safety and reliability through features such as increased hardware reliability and stability, and improved failure detection capability. It however makes the systems and their safety analysis more complex. Originally, safety analysis was applied to hardware system components and formal methods mainly to software. For software-controlled or digitalized systems, it is necessary to integrate both [1].

Fault tree analysis (FTA) which has been one of the most widely used safety analysis technique in nuclear industry suffers from several drawbacks as described in [2]. In this work, to resolve the problems, FTA and model checking are integrated to provide formal, automated and qualitative assistance to informal and/or quantitative safety analysis. Our approach proposes to build a formal model of the system together with fault trees. We introduce several temporal gates based on timed computational tree logic (TCTL) to capture absolute time behaviors of the system and to give concrete semantics to fault tree gates to reduce errors during the analysis, and use model checking technique to automate the reasoning process of FTA.

2. Background

In this section, some of the techniques used in this work such as model checking, temporal fault tree analysis and so on are described.

2.1 Model Checking and UPPAAL

Model checking is the most usual formal verification technique and a proven-effective and automated technique in verifying complex behavior of concurrent systems. A model checker, given the system description and property specification, determines if the properties hold in the model or not. Among several model checkers being used in industry, we selected a real time model checker UPPAAL [3] to support our approach because it supports elaborate verification of time-related system behavior with friendly graphic user interface. The model checker will either terminate with the answer true, indicating that the system model satisfies the property, or false, indicating that the system model dose not satisfy the property and provides a counter example that shows an execution trace that violates the property. The counter example is one of the most useful features of model checking, as it allows users to quickly understand why a property is not satisfied.

2.2 Temporal Fault Tree

We developed several new temporal gates based on TCTL to describe dynamic behaviors of system and defined the temporal gates, some dynamic gates and static gates in the previous work [2]. With these gates we easily specify the temporal dependence between events and preserve the simple, qualitative and visual nature of the fault trees. Each temporal gate has it is own usage. For example, the 'continuity gate' is useful in the description of the situation where an event should continue for at least particular time after the other event has occurred and the corresponding expression in the form of TCTL is $EG[\varphi \rightarrow AG_{\leq \alpha}\ \psi]$ (ψ continues for at least α time units after φ has occurred). Users could easily understand the usages of other temporal gates from the intuitive meaning in the definition of gates in [2].

3. Case Study

We applied our approach to digital feedwater control system (DFWCS) which is the benchmark system in [4]

3.1 DFWCS Modeling in UPPAAL

In UPPAAL, a system is modeled as a network of several such timed automata in parallel. The model is further extended with bounded discrete variables that are part of the state. These variables are used as in programming languages: they are read, written, and are subject to common arithmetic operations. A state of the system is defined by the locations of all automata, the clock constraints, and the values of the discrete variables. Every automaton may fire an edge (sometimes misleadingly called a transition) separately or synchronize with another automaton, which leads to a new state. We made fifteen separated models (which called 'template' in UPPAAL) for describing DFWCS behavior. Some of the templates are for describing corresponding components behaviors of DFWCS, and others are additional templates for describing special dependency between components, for example, MFVC_PDI_Down template for hot spare dependency between MFV controller and PDI.

3.2 Model Checking and Results

We made a fault tree of DFWCS based on the system description and FMEA results in [4]. All the information of a fault tree is translated into UPPAAL query language for automatic verification. First, fault tree gates are translated to corresponding UPPAAL query language based on transition rules between TCTL and CTL, and UPPAAL query language. After completion of the translation process, the translated fault tree information which is now system property leading to unintended system state (hazardous state) is verified by UPPAAL model checker against UPPAAL system model implemented previously.

Through the verification process, we can verify the property that 'when MC watchdog timer is not reset exactly t0 time units after watchdog timer started and BC fails, if an operator can not intervene within t2 time, DFWCS control fails at one time.

But the verification result of the property, E<>(Sensors.InV_Soutputs imply MC.Down) is 'the property is not satisfied'. This is because DFWCS has a kind of fault tolerant function where, even though invalid inputs from sensors are inserted, computers do not fail immediately and wait for one processing time after invalid inputs from the sensors. We can conclude more easily that the fault tree has flaws, and hence the analysis result is also erroneous. The corrected fault tree is shown in Figure 1. In the corrected fault tree, 'Promptness gate' is used to describe temporal dependency between invalid sensor inputs and MC failure because it is useful in the description of the situation where an event occurs within particular time after the other event has occurred. Therefore, with this gate, we can describe the event that if valid inputs from sensors are not inserted to computers within t1 (which is one processing time) time unit after invalid inputs from sensors, MC is down (or failed).

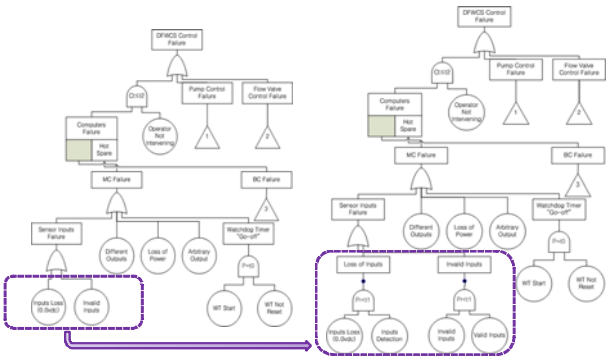

Fig. 1. Corrected fault tree of the partial fault tree

4. Conclusions

This paper demonstrated that the new temporal gates are useful to capture dynamic behaviors of system precisely and that model checking technique is helpful when we validate the correctness of informal safety analysis such as FTA. Our approach not only formalizes the semantics of fault trees, but it also extends the expressive power of FTA to model temporal ordering of events. But the concept will need further improvements and validation by larger scaled case studies. The gates proposed here are not yet sufficient to model all situations that arise in digitalized systems. Thus we intend to add some new gates to our framework, if necessary. In other for our method to have any strength over other method reviewed in this paper, the supporting tool to automate the proposed method should be developed.

REFERENCES

[1] N.G. Leveson, Safeware: system safety and computers, Addison- Wesley, New York, 1995.

[2] Kwang Yong Koh and Poong Hyun Seong, SA**CS**^{2}: A Dynamic and Formal Approach to Safety Analysis for Complex Safety Critical System, Proceedings of Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT-2009), Knoxville, Tennessee, April 5-9, 2009.

[3] B.Berard and et al., Systems and Software Verification Model-Checking Techniques and Tools, Springer-Verlag, Berlin Heidelberg, 2001.

[4] T. Aldemir, M.P. and et al., Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments, NUREG/CR-6942, U.S. NRC, Washington, D.C., 2007.