# NuFTA: A CASE Tool for Automatic Software Fault Tree Analysis

Sanghyun Yun*, Dong-Ah Lee, Junbeom Yoo
*Division of Computer Science and Engineering, Konkuk University*
*1 Hwayang-dong, Gwangjin-gu, Seoul, 143-701, Republic of Korea*
** Corresponding author: pctkdgus@konkuk.ac.kr*

## 1. Introduction

Software fault tree analysis (SFTA) [1] is widely used for analyzing software requiring high-reliability. In SFTA, experts predict failures of system through HAZOP (Hazard and Operability study) or FMEA (Failure Mode and Effects Analysis) and draw software fault trees about the failures. Quality and cost of the software fault tree, therefore, depend on knowledge and experience of the experts. This paper proposes a CASE tool NuFTA in order to assist experts of safety analysis. The NuFTA automatically generate software fault trees from NuSCR formal requirements specification [2]. NuSCR is a formal specification language used for specifying software requirements of KNICS RPS (Reactor Protection System) in Korea. We used the SFTA templates proposed by [3] in order to generate SFTA automatically. The NuFTA also generates logical formulae summarizing the failure's cause, and we have a plan to use the formulae usefully through formal verification techniques.

## 2. **Background**

### *2.1 NuSCR*

NuSCR [2] is a formal software requirements specification method for digital nuclear plants protection systems. Figure 1 depicts a simplified example of FOD (Function Overview Diagram) for the KNICS RPS BP (Bistable Processor). FOD is one of basic constructs of NuSCR. SFTA for NuSCR specifications using NuFTA starts when a failure node is defined in the FOD. In the SFTA of RPS, a failure may be the case which makes the system stop (shutdown: "*th_Trip =1*") abnormally. The "*th_Trip*" node in Figure 1 becomes a failure node for the SFTA.

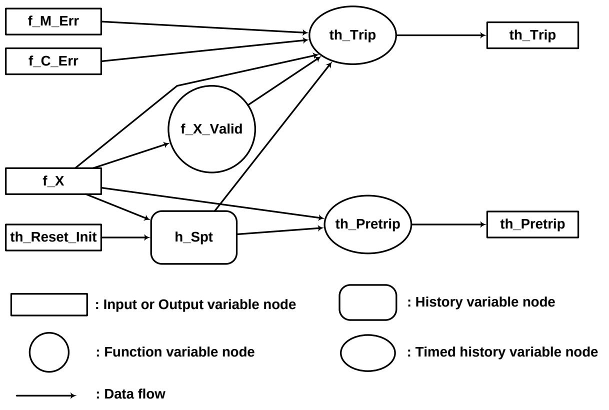

Fig. 1. A simplified example of NuSCR FOD

NuSCR has three types of nodes within FODs, as depicted in the legend in Figure 1.They all have own SFTA templates and associated algorithm for generating them respectively.

### *2.2 NuSCR SFTA Templates*

The SFTA method [3] for NuSCR suggests 3 types SFTA templates corresponding three NuSCR nodes. In figure 2, the top event indicates a failure (the '*output*' is equal to the '*value*') and the branches outgoing from the OR-gate denote conditions generating the '*value*'. The left two branches represent the cases which the value causing the top failure is the same as the right hand side of action statement, and shared/included by, respectively. The rightmost branch describes that the relationship between them cannot be decided. The templates for history variable and timed-history variable nodes have one additional branch in order to analyze state-related conditions. [3] suggests an extended FSM in order to unfold the states and value information, and we modified and developed the algorithm of constructing extended FSMs.

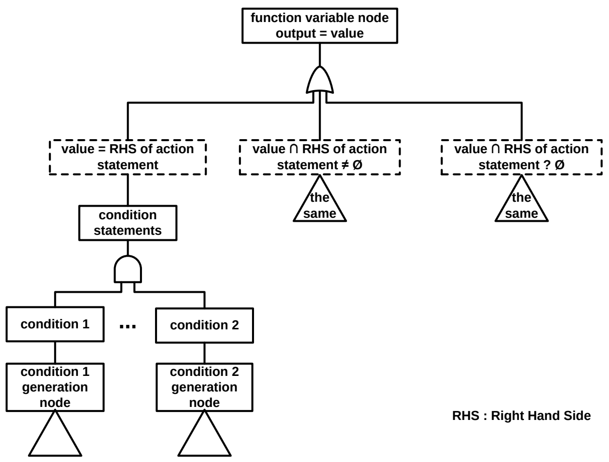
function variable node
output = value
value = RHS of action statement
value ∩ RHS of action statement ≠ Ø
value ∩ RHS of action statement ? Ø
the same
the same
condition statements
condition 1
...
condition 2
condition 1 generation node
condition 2 generation node
RHS : Right Hand Side

Fig. 2. The SFTA template for function variable node

## 3. **NuFTA**

We developed the NuFTA (ver. 0.8) in Java. It automatically generates a software fault tree from the requirements specification for KNICS RPS BP written in NuSCR formal specification method. It receives an NuSCR specification in XML from NuSRS (ver 2.0) [4], generates a software fault tree and store it in XML. It also produces a logic formula summarizing the software fault tree. The SFTA procedure using the NuFTA is summarized as follows:

① The NuFTA reads a NuSCR specification and checks all contents for guaranteeing integrity of the specification

② The NuFTA identifies one node which generates the failure case as its output.

③ As shown figure 3, the NuFTA finds causes of output by tracing back FOD to the node's input and extends software fault tree with corresponding NuSCR FTA templates. The NuFTA defines values of input variable nodes as basic events, states as undeveloped events.

④ Until all leaf nodes are basic events or undeveloped events, continue on step 3.

⑤ The NuFTA generates logical formula from software fault tree using the method [5] which interprets software fault tree to logical formula.

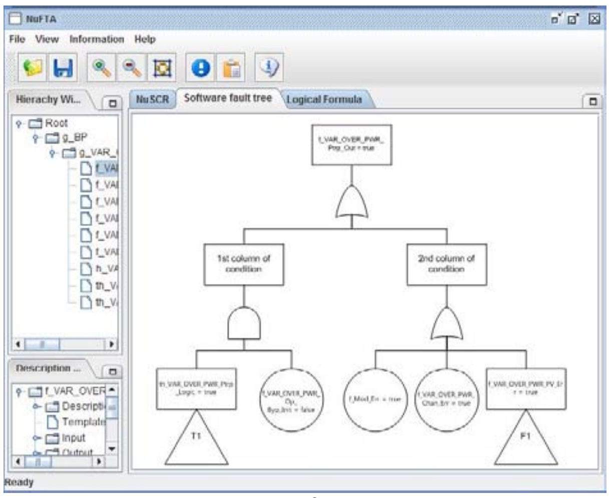

Fig 3. A screen-dump of the NuFTA (ver 0.8)

## 4. Conclusion and Future Work

The paper suggested a CASE tool NuFTA, an assistant for generating software fault trees from NuSCR formal requirement specification. It can decrease time spent for constructing software fault trees manually and reduce cost for analyzing using it. We are currently focusing on integrating the NuFTA into the software verification and validation framework we proposed in [4].

**Acknowledgement**
This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency (NIPA-2010-(C1090-0903-0004)

## REFERENCES

[1] N. G. Leveson and P. R. Harvey, Software fault tree analysis, Journal of Systems and Software, Vol. 3, pp. 173–181, 1983.

[2] Jumbeom Yoo, Taihyo Kim, Sumgdeok Cha, Jangsu Lee, and Han Seong Son, A Formal Software Requirements Specification Method for Digital Nuclear Plants Protection Systems, Journal of Systems and Software, Vol.74, No.1, pp. 73-83, 2005.

[3] T. Kim, Property-based Theorem Proving and Template-based Fault Tree Analysis of NuSCR Requirements Specification, Ph.D Dissertation, KAIST, 2005.

[4] Junbeom Yoo, Eunkyoung Jee, Sungdeok (Steve) Cha, Formal Modeling and Verification of Safety-Critical Software, IEEE Software, Vol.26, No.3, pp.42-49, May/June 2009.

[5] Hansen, K. M., Ravn, A. P., and Stavridou, V. From safety analysis to software requirements, IEEE Transactions on Software Engineering July 1998; 24(7):573–584.