Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 7-8, 2026

Comparative Analysis of PSA Workflows in SAPHIRE and AIMS-PSA: Impact on Quantification Results

A Ra Lee [a], Man Cheol Kim [a*]
[a, a*] *Department of Energy Systems Engineering, Chung-Ang University, 84 Heukseokro, Dongjak-gu, Seoul 06974*
[*] *Corresponding author: charleskim@cau.ac.kr*

Keywords : PSA, SAPHIRE, Process Flags, Logical Loops, Time-Dependent Events

1. Introduction

Probabilistic safety assessment (PSA) is a key methodology for nuclear power plants (NPPs) that structures accident scenarios using event trees (ETs) and fault trees (FTs) and quantifies the core damage frequency (CDF) by generating and quantifying minimal cut sets (MCSs). This quantification is primarily performed using PSA software tools such as SAPHIRE (Systems Analysis Programs for Hands-on Integrated Reliability Evaluations), developed by the Idaho National Laboratory (INL) for the U.S. Nuclear Regulatory Commission (NRC), and AIMS-PSA (Advanced Information Management System for Probabilistic Safety Assessment), developed by the Korea Atomic Energy Research Institute (KAERI). Although both tools share a common ET/FT-based quantification framework, they differ in processing approaches: SAPHIRE provides analyst-configurable options (e.g., Process Flags) that allow flexible control of sequence development, whereas AIMS-PSA performs cut set generation and quantification through an engine-centered workflow based on Fault tree reliability evaluation expert (FTREX). These workflow and engine-behavior differences can affect cut set composition even for the same input model, potentially leading to discrepancies in quantification results.

Accordingly, this study compares the practical modeling behavior of SAPHIRE and AIMS-PSA focusing on three mechanisms that directly affect quantification outcomes: (1) sequence development rules controlled by Process Flags, (2) logical loop handling (loop-free transformation versus engine-level resolution), and (3) convolution-based time-dependent event quantification to complement limitations of static logic models. The results provide a technical basis for improving reproducibility in both PSA tool application and tool switching, and for establishing configuration strategies aligned with analysis objectives.

2. Methods and Results

For a fair comparison, the same ET/FT logic structure and basic-event parameters were implemented in SAPHIRE and AIMS-PSA, and quantification was performed using the rare event approximation (REA). Sections 2.1–2.3 compare tool behaviors for each mechanism, quantify their impacts on cut set composition and quantification results, and summarize practical implications, including tool-specific constraints and workarounds.

2.1 Process flags / Sequence development rules

This subsection examines how sequence development rules at the sequence cut set generation stage influence quantification results. In SAPHIRE, Process Flags are analyst-specified identifiers that determine how top events are treated when event tree sequences and associated fault tree logic are solved. SAPHIRE allows analysts to select success and failure branch handling and the top-event expansion level via Process Flags, whereas AIMS-PSA follows an engine-centered default workflow based on FTREX.

Sequence development rules determine the balance between accuracy and computational burden. Because expansion of complementary logic for success branches can rapidly increase the number of cut sets, both tools commonly rely on delete-term approximation (DTA) for efficiency. According to Jung (2009) [1], DTA is a logical operation that removes impossible cut sets that arise when success-state events are included in failure cut sets.

The impact of rule selection can be summarized by two representative structures. First, when non-rare events appear in a success branch such that the reduction in success probability (1-P) is non-negligible, a DTA-based default rule may not fully reflect the probabilistic contribution of success conditions and can be conservatively biased. To address this, the I flag (success-logic preservation) explicitly includes success conditions in the cut sets, thereby reducing this conservatism. Second, for dependency structures where top events share a common event, retaining FT expansion preserves correlation, whereas the Y flag (developed event replacement) can remove dependency information and implicitly introduce an independence assumption, potentially leading to non-conservative underestimation.

Therefore, Process Flags provide a strategic means to adjust expansion rules according to model characteristics and analysis objectives. Analysts should diagnose non-rare success-branch conditions and shared-event dependencies before applying simplifications and establish tool settings that best fit the intended purpose of the analysis.

2.2 Logical loop resolution

This subsection compares loop detection and quantification workflows across PSA tools when inter-system interdependencies appear as logical loops after ET/FT aggregation. As a case study, the Total Loss of Component Cooling Water (TLOCCW) event in the KAERI Pilot Plant PRA model was selected. The model contains a closed-loop dependency structure, such as Component Cooling Water (CCW) → Electrical Power System (EPS) → CCW, reflecting the interlinked nature of support-system electrical power and cooling water supply.

When loops were present, the two tools showed different workflows. SAPHIRE detects loops during model integration and provides the warning message "Risk error of Logic Loop Discovered" along with loop-path information; if circular logic is present, the procedure is interrupted at the cut set generation stage. Accordingly, quantification in SAPHIRE was performed after converting the model into a loop-free structure using the analytical loop-breaking procedure proposed by Jung (2005) [2]. In contrast, quantification using the FTREX engine within AIMS-PSA was completed for the same looped model without additional manual structural modification, indicating that FTREX supports quantification for models containing logical loops.

The TLOCCW frequency after loop removal was $2.120 \times 10^{-6}\, yr^{-1}$ (28 MCS), whereas FTREX-based quantification within AIMS-PSA yielded $2.123 \times 10^{-6}\, yr^{-1}$ (28 MCS) with the loop retained. In this case, differences in loop-resolution workflows affected quantifiability and analyst intervention more directly than the final risk value.

2.3 Time-dependent event modeling

This subsection considers a station blackout (SBO) scenario initiated by a loss of offsite power (LOOP), in which the first emergency diesel generator (EDG) fails to run, the second EDG subsequently fails to run, and offsite AC power is not recovered before core uncovery after failure of the second EDG. For this scenario, in the conventional static PSA modeling approach used here in AIMS-PSA, relevant events are treated as independent, and the sequence probability is evaluated by direct multiplication of the corresponding event probabilities. In contrast, SAPHIRE applies convolution-based time-dependent quantification, in which the same SBO sequence is modeled according to the temporal order of the two EDG fail-to-run events and offsite power recovery. In contrast to the static approximation, core cooling is assumed to be lost only after the second EDG fails, which increases the time available to recover offsite power and reduces the sequence probability. This interpretation is consistent with Kim (2024) [3], who showed that the conventional ET/FT approach can be conservative for SBO scenarios when fail-to-start and fail-to-run failures are not distinguished and the time available to recover offsite power is not explicitly represented.

For SBO quantification, offsite power recovery was modeled using the switchyard-centered LOOP recovery data from NUREG/CR-6890 (2020 Update) [4], while the EDG fail-to-run failure rate was taken from Johnson and Schroeder [5]. The mission time and the core uncovery criterion were specified as assumed values.

The static results were similar (AIMS-PSA: 2.048×10^{-5} ; SAPHIRE: 2.005×10^{-5}). When time dependence was incorporated, SAPHIRE agreed closely with MATLAB numerical results (SAPHIRE: 4.084×10^{-6}; MATLAB: 4.146×10^{-6}), yielding a sequence probability approximately five times lower than the static approximation. These results demonstrate that static ET/FT quantification can be conservative for SBO scenarios, whereas convolution-based time-dependent quantification yields more realistic estimates by capturing how component fail-to-run timing affects the time available to recover offsite power.

3. Conclusions

This study analyzed how differences in quantification workflows and engine behavior in SAPHIRE and in the AIMS-PSA workflow considered in this study affect PSA quantification results. The results show that differences in sequence development rules can shift the direction of conservatism depending on whether non-rare events are included in success branches and whether dependency structures exist. For logical loops, the final results were nearly identical, but the workflows differed in terms of required analyst intervention. For the SBO scenario, PSA quantification differed depending on how the temporal order of fail-to-run events and offsite AC power recovery was modeled, with the convolution-based treatment accounting for the time available to recover offsite power. Overall, reproducible PSA quantification requires attention not only to the underlying model but also to tool-specific settings and workflows, and these comparisons provide practical guidance for configuration selection.

REFERENCES

[1] W. S. Jung, ZBDD algorithm features for an efficient Probabilistic Safety Assessment, Nuclear Engineering and Design, Vol. 239, pp. 2085-2092, 2009.

[2] W. S. Jung and S. H. Han, Development of an analytical method to break logical loops at the system level, Reliability Engineering and System Safety, Vol. 90, pp. 37-44, 2005.

[3] M. C. Kim, Comparison of event tree/fault tree and convolution approaches in calculating station blackout risk in a nuclear power plant, Nuclear Engineering and Technology, Vol. 56, pp. 141-146, 2024.

[4] N. Johnson and Z. Ma, Analysis of Loss-of-Offsite-Power Events: 2020 Update, INL/EXT-21-64151, Idaho National Laboratory, Nov. 2021.

[5] N. Johnson and J. Schroeder, Analysis of Loss-of-Offsite Power Events 1987–2015, INL/EXT-16-39575, Idaho National Laboratory, July 2016.