

Practical Challenges in Implementing Semi-Detailed Fire Scenarios within Existing Fire Probabilistic Safety Assessment Software Tools

Kyungho Jin, Yong Hun Jung*, Dae Il Kang
Korea Atomic Energy Research Institute, Daejeon, Republic of Korea

*jungyh@kaeri.re.kr

***Keywords:** Semi-Detailed Fire Scenarios, Fire Probabilistic Safety Assessment, AIMS-PSA

1. Introduction

One of the key elements of fire probabilistic safety assessment (PSA) is the development of fire scenarios. As the fire scenarios need to comprehensively account for ignition sources, targets (e.g., cables), fire protection features and propagation, the number of scenarios to be addressed becomes substantial. As a result, considering the combinations between every ignition source and potentially affected targets — for example, through detailed fire modeling — is practically infeasible. Consequently, the fire scenarios within a compartment are generally collapsed into a single *bounding case*, in which any fire is assumed to result in full-room burnout (FRB).

While such a bounding scenario effectively reduces the analytical burden, the underlying conservative assumptions can make the PSA results excessively conservative. Given the growing emphasis on PSA quality in support of risk-informed applications, undue conservatism can distort PSA results and limit their usability in risk-informed decision-making.

To balance the level of detail against the analytical burden, the Korea Atomic Energy Research Institute (KAERI) has proposed a semi-detailed approach in which the bounding case is subdivided into several distinct sub-scenarios, with fire modeling performed for each sub-scenario using a semi-detailed method [1].

This multiple fire scenario approach combined with semi-detailed fire modeling is hereafter referred to as the *semi-detailed fire scenario* framework. This framework is intended to provide a more realistic representation of fire scenarios than the conventional bounding treatment. However, methodological feasibility does not necessarily guarantee that this framework can be readily implemented within existing fire PSA software tools (e.g., AIMS-PSA [2]).

Accordingly, this paper identifies the practical challenges encountered in applying such tools to the semi-detailed fire scenario framework and discusses how these challenges might be addressed.

2. Semi-Detailed Fire Scenario Framework

2.1 Overview of Semi-Detailed Fire Scenario Framework [1]

In the conventional bounding scenario, every ignition source within a fire compartment is assumed to lead to FRB (i.e., any fire within a compartment is presumed to damage all target items present); consequently, the ignition frequencies of all sources can be aggregated and considered a single scenario. Under the conventional FRB assumption, the severity factor (SF) and non-suppression probability (NSP) are inevitably set to unity, so that the entire fire ignition frequency is allocated to the worst-case conditional core damage probability (CCDP).

To reduce this conservatism, the semi-detailed framework divides a single FRB scenario into multiple mutually exclusive fire damage states (FDS) distinguished by the extent of damage and the timing of suppression [1]. In the form proposed in [1], a single FRB scenario is subdivided into three sub-scenarios, as follows:

- *FDS0*: Only the ignition source itself is damaged, either because the fire is not severe enough to damage the first target (e.g., the nearest and most vulnerable target within the compartment) or because it is suppressed beforehand.
- *FDS1*: The fire damages the first target but is suppressed before the remaining targets in the compartment are affected.
- *FDS2*: The fire is not suppressed and damage extends to all targets within the compartment, corresponding to the damage target sets assumed in FRB.

Since the framework requires only the identification of the first target, the additional analytical effort remains modest.

2.2 Scenario Proliferation in the Semi-Detailed Framework

Whereas the conventional approach aggregates the ignition frequencies of all sources within a compartment into a single bounding scenario, the semi-detailed

framework requires three sub-scenarios for each ignition source, so that a compartment containing n ignition sources yields $3n$ scenarios. Consequently, the total number of fire scenarios increases in proportion to the number of ignition sources across all fire compartments. Table I compares the number of fire scenarios generated under the bounding and semi-detailed approaches for the reference nuclear power plant analyzed in this study.

Table I: Comparison of the number of fire scenarios between bounding and semi-detailed approach

	Bounding case	Semi-detailed case
No. fire compartments	221	
No. ignition sources for all compartments	2,507	
No. fire scenario ^{*,**}	221	7,521

* Fire propagation scenario is excluded for computational simplicity.
** In practice, expert judgment is applied to determine which of the scenarios are retained for analysis. The comparison presented here is based on the number of scenarios generated arithmetically, without such screening.

3. Implementing the Semi-Detailed Framework in Existing Fire PSA Software Tools

3.1 Practical Challenges in Addressing a Large Number of Fire Scenarios

A fire PSA involves several software tools, each supporting a different stage of the analysis. This paper focuses particularly on AIMS-PSA [2], because most of the difficulties stemming from the increased number of fire scenarios arise in the model building and quantification stages, which are carried out with this tool.

AIMS-PSA provides a script-based function called SIMA (Script Interpreter for Mapping Algorithm) [2], which allows the fault tree structure to be modified as required for a fire PSA. In practice, SIMA is used to adapt a base fault tree model to each fire scenario — for example, by adding house events at the appropriate locations to consider the equipment affected by the fire. SIMA operates through text-based commands that control the existing fault tree. Table II shows examples of commands in a SIMA script.

Table II: Examples of commands in a SIMA script

```
;1 (1) : %ILOCV %F-FCxxx-xxxA ZDE-3xxxVxxx1
;[Initiating Event]
;SeverityProba=1, NonSupProba=1
Add+ %ILOCV %F-FCxxx-xxxA
Value %F-FCxxx-xxxA 1.03E-05
Desc %F-FCxxx-xxxA fire initiating event ...

;[Component Failure]
; SeverityProba=1, NonSupProba=1, CondProba_Final=1
Add+ ZDE-3xxxVxxx1 G&F-[F2]

;2 (2) : %ILOCV %F-FCxxx-xxxA ZDE-3xxxVxxx2
;[Component Failure]
;SeverityProba=1, NonSupProba=1, CondProba_Final=1
Add+ ZDE-3xxxVxxx2 G&F-[F3]
:
```

While this command-based approach works well for a limited number of scenarios, it becomes impractical as the script length increases. Even a single fire scenario may require tens to hundreds of SIMA commands, depending on how the scenario is defined. Under the conventional bounding approach, only one scenario per compartment needed to be described in this way, and the command set remained manageable.

In the semi-detailed framework, however, the increased number of scenarios and the number of commands per scenario compound one another, causing the total volume of SIMA commands to quickly exceed what AIMS-PSA can process within a practical time frame. Table III compares the number of SIMA commands required between the two approaches.

Table III: Comparison* of the number of SIMA script commands between the bounding and semi-detailed fire scenario (FDS2 only)

	Bounding case	Semi-detailed case**
No. SIMA script commands	5,807	124,562
Data volume (MB)	1.5MB	32.2MB

* This comparison covers only the fire compartments within Primary Auxiliary Building (PAB)

** For FDS2 only. These assume the same damage target sets as the bounding case but are generated separately for each ignition source.

It should be noted that the results in Table III compare SIMA commands generated only for the FDS2 scenarios, which correspond to the same damage target sets as the bounding case, so that the two approaches can be compared on an equivalent basis. If a SIMA script were generated for the full set of sub-scenarios, this burden would become considerably more severe.

Even at this reduced scope, the increased volume of input becomes apparent at three stages of the AIMS-PSA workflow: one-top build, quantification, and review of results.

The first is the one-top build, in which the SIMA commands are applied to the base fault tree to construct a single integrated fault tree covering all fire scenarios. Because every SIMA command must be processed sequentially rather than in parallel, the build time grows rapidly as the command set expands. For example, building the one-top model using the SIMA scripts listed in Table III took approximately 30 seconds under the bounding approach, compared with about 32 minutes under the semi-detailed framework. It should be noted that this comparison considers only the PAB and FDS2 conditions; the gap would widen further if FDS0 and FDS1, or fire propagation scenarios, were included.

The second is quantification. As noted above, the one-top model built under the semi-detailed framework is substantially larger than that of a bounding case, and the quantification time is also expected to differ accordingly. Generating minimal cut sets from such a model requires a computation time that is difficult to accommodate in practice.

The third is the review of results. Once quantification is completed, the number of minimal cut sets produced is so large that displaying and reviewing them becomes impractical.

3.2 Possible Approaches to Addressing the Identified Challenges

The challenges described above stem from a common cause: the volume of input (i.e., the commands in the SIMA script) that must be processed in a single run is too large. Within the current software environment, the problem may therefore be mitigated by dividing the workload.

Although AIMS-PSA does not provide a dedicated parallelization function, multiple AIMS-PSA projects can be run concurrently as independent instances, with the available CPU cores allocated automatically. Accordingly, the workload can be partitioned in practice without modifying the tool. Since semi-detailed fire scenarios are mutually exclusive, they carry no logical dependency on one another and can therefore be quantified independently without affecting the results. Therefore, the SIMA script can be partitioned into k subsets, each assigned to a separate project, so that k one-top fault trees are built and quantified independently rather than as a single large model.

Because the command set handled in each project is reduced to approximately $1/k$ of the original, the build and quantification times per project fall to a level that can be accommodated in practice, and the number of minimal cut sets returned by each project remains small enough to display and review. Fig. 1 shows an example of workload distribution across multiple AIMS-PSA projects, with the SIMA script partitioned into 10 subsets.

This approach does not reduce the total computational effort, but it distributes the computation into manageable units and thereby avoids the limits encountered when the entire model is processed at once. It does, however, require abandoning the construction of a single integrated one-top model, and it also calls for an additional capability to merge the k sets of minimal cut sets into a unified result.

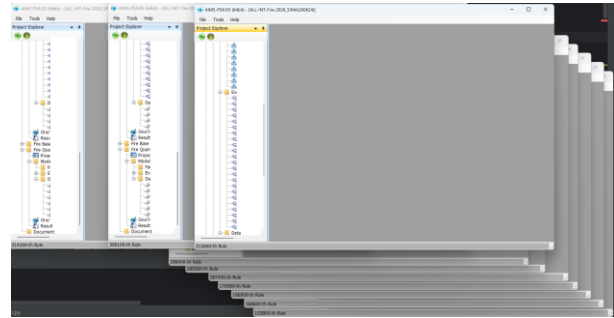


Fig. 1. Example of SIMA script partitioned into 10 subsets and distributed across multiple AIMS-PSA projects (One-top build)

4. Conclusion

This paper examined the practical challenges of implementing the semi-detailed fire scenario framework within existing fire PSA software tools. Because the framework replaces a single bounding scenario with multiple sub-scenarios per ignition source, the volume of SIMA script to be processed in AIMS-PSA increases sharply, and the resulting burden appears at the one-top build, quantification, and result review stages.

Partitioning the script across multiple concurrently executed projects mitigates this burden within the current software environment, but it forgoes a single integrated one-top model and requires an additional capability to merge the resulting minimal cut sets. The effectiveness of this partitioning approach in reducing computation time needs to be verified through further study.

In the short term, the challenges can be managed by partitioning the script across multiple projects. In the longer term, however, routine application of the semi-detailed framework will require further development of the supporting software tools.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (RS-2022-00144204).

REFERENCES

- [1] Y. Jung, K. Jin, and D. Kang, "A Semi-Detailed Method for Fire Scenario Analysis in Fire PSA," *Transactions of the Korean Nuclear Society Autumn Meeting*, Changwon, Korea, Oct. 30-31, 2025
- [2] KAERI, *AIMS-PSA Release 2 Reference Manual*, Rev.1.2, 2015