

Development of DICE–Flownex for Dynamic Accident Sequence Analysis of SMRs

Yuntae Gwak^a, Yongjoon Lee^a, Gyunyoung Heo^{*a}

^a Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Yongin, Republic of Korea

*Corresponding author: gheo@khu.ac.kr

Abstract: Conventional probabilistic safety assessment (PSA) evaluates risk by combining static event trees and fault trees, but it requires the analyst to enumerate the accident sequences in advance. This is a particular limitation for small modular reactors (SMRs), whose integral and passive designs give rise to accident sequences and success criteria that are not yet well established. The dynamic event tree (DET) method addresses this by generating branches automatically from a time-dependent simulation, so that accident sequences are discovered rather than presumed. The Dynamic Integrated Consequence Evaluation (DICE) platform supports such DET analyses by coupling an external deterministic code as its Physics module, and it has previously been coupled with MARS-KS and MELCOR. In this study, DICE was coupled with the thermal-hydraulic system analysis code Flownex to supply the realistic, system-level physics that automated sequence discovery requires. Flownex models a plant as a network of components with integrated control logic and can represent coupled power, thermal, and fluid systems, which makes it well suited to SMRs and to integrated energy and Power-to-X (P2X) applications. A data-exchange interface was designed using the Flownex Python API to keep the variables and actuation signals of the two codes synchronized at every time step, generalized as a Python coupling layer so that other system codes can be coupled with little change to DICE. The coupled system was demonstrated on a loss-of-coolant accident (LOCA) in a hypothetical SMR, in which the DET automatically enumerated the accident sequences arising from the timing and combination of safety-function demands.

Keywords: Dynamic PSA, DICE, Flownex, SMR, Accident Sequence Analysis, Code Coupling

1. INTRODUCTION

Probabilistic Safety Assessment (PSA) has been the standard methodology for risk evaluation of nuclear power plants since WASH-1400 (U.S. NRC, 1975) established the framework combining the event tree (ET) and the fault tree (FT) [1]. The 1979 TMI accident, however, showed that a static ET does not adequately capture the state transitions that arise as operator interventions and system failures interact over time, and it has been noted that the conservative assumptions commonly used in event trees do not always lead to conservative risk estimates [2]. Rather than replacing the well-established static ET framework, these observations point to the value of a complementary approach that accounts for time-dependent interactions, especially for analyses with a large diversity of scenarios such as those of advanced reactor designs.

The dynamic event tree (DET) method was developed for this purpose. DET builds the event tree automatically on a simulation basis, coupling deterministic thermal-hydraulic analysis with probabilistic models in the time domain so that multi-phenomena interactions can be analyzed [3]. Because branching is driven by the simulated system state rather than by a pre-specified sequence list, the DET represents time-dependent branching and the variability of operator actions explicitly, and it discovers accident sequences that a static ET, which relies on sequences enumerated in advance by the analyst, may miss.

This capability is especially valuable for new reactors, whose accident sequences and success criteria are not yet as well established as those of large light-water reactors, so a method that enumerates sequences automatically is well matched to the problem. DICE (Dynamic Integrated Consequence Evaluation), developed at Kyung Hee University, is a platform built to support such DET analyses [4]. It adopts a plug-in structure that couples an external deterministic simulation code as its Physics module, and it has been coupled with safety analysis codes such as MARS-KS and MELCOR for Level 1 and Level 2 PSA [3].

The purpose of this study is to give DICE the realistic, system-level thermal-hydraulic physics that automated sequence discovery for SMRs requires, by coupling it with the thermal-hydraulic system analysis code Flownex. Flownex represents a plant as a network of components with integrated control logic and can model multi-module systems and coupled power, thermal, and fluid systems; these features suit advanced reactors such as SMRs as well as integrated energy systems and Power-to-X (P2X) analyses. Through the DICE–Flownex coupling, dynamic accident-sequence analysis that reflects system-level behavior becomes possible beyond the scope of conventional safety analysis codes.

To this end, a data-exchange structure between the Flownex API and DICE was designed, keeping the variables and states of both codes synchronized at every time step. The interface is implemented as a Python coupling layer that generalizes coupling with external codes, so that new analysis modules can be added easily and the reliability-evaluation logic can be modified by the user at the Python level. The coupled system was then applied to a LOCA in a hypothetical SMR, confirming that system-level simulation results can be incorporated into dynamic accident-progression analysis.

2. PLATFORM OVERVIEW

2.1. Architecture and Modules

DICE is a DET analysis tool developed at Kyung Hee University. DICE consists of four main modules, namely the Scheduler, Physics, Diagnosis, and Reliability modules, with the Scheduler coordinating the inputs and outputs of the other three modules from a central position.

The Physics module drives the deterministic simulation code to compute the time-dependent thermal-hydraulic behavior. The Diagnosis module checks set-point and procedure rules against the monitoring variables and determines branching from automatic and manual signals, and the Reliability module evaluates the component failure states and branch probabilities for the resulting candidates. The Scheduler calls these three modules in sequence and manages the branch data. Because DICE uses a plug-in architecture with a separate DLL per module, the user can replace the Physics code or the Diagnosis algorithm according to the analysis objective.

DICE provides two analysis modes. In the DDET (Discrete Dynamic Event Tree) mode, all predefined branches are generated at each set point following the MECE principle and their probabilities are quantified with fault-tree techniques. In the MCET (Monte Carlo Event Tree) mode, component failures may occur at arbitrary times and a single sequence is advanced by random-number comparison, which suits the exploration of unidentified scenarios.

2.2. Previous Achievements

Various thermal-hydraulic codes can be coupled to the DICE Physics module depending on the analysis objective. To date, MARS-KS 1.5 and MELCOR 2.x have been coupled and used for Level 1 and Level 2 PSA analyses [4].

The MARS-KS coupling uses the Interactive Variable feature (the 800-series cards): the monitoring variables and the variables that trigger the Trip cards are defined as Control Variables and registered as Interactive Variables, so they can be exchanged with an external program in real time.

The MELCOR coupling uses the Advanced Communication Facility (ACF) of MELCOR 2.x. By registering the monitoring and control variables with the ACF through User-Defined Functions and the `dynamic_link` feature of the CF Package, DICE can read MELCOR's state and update its inputs externally.

Both cases yielded the same interface requirements: a real-time variable-exchange channel between the external code and DICE, a means of controlling the trip logic externally, and synchronization on a per-time-step basis. These requirements are the starting point for the Flownex coupling introduced in this study.

3. DICE–FLOWNEX INTERFACE DESIGN

DICE includes a Physics module that computes the system's thermal-hydraulic behavior, and this module is a slot into which an external analysis code can be plugged. The coupling developed here places Flownex into that slot: Flownex does not run as a separate code beside DICE but serves as DICE's physics-calculation engine, supplying the system states needed to advance an accident scenario. DICE's Scheduler governs the analysis, at every time step delegating the system calculation to Flownex and then using the Diagnosis and Reliability modules to determine the branching. The two codes are interlocked within a single process through the Python interface provided by Flownex, and the exchanged variables and actuation signals are defined in a separate configuration file rather than in the code, so neither the other DICE modules nor the internals of the Flownex model need to be modified.

3.1. Flownex Overview and Motivation for the Coupling

Flownex is a one-dimensional thermo-fluid system analysis code that represents a system as a network of components such as pipes, pumps, valves, tanks, and heat exchangers connected by nodes and elements. It solves the mass, momentum, and energy conservation equations in both steady-state and transient conditions and can simultaneously treat compressible, incompressible, and two-phase fluids, heat conduction through solid structures, and system control logic. It also provides nuclear-specific components with neutron kinetics, so that core power and system thermal-hydraulics can be analyzed together within a single model.

These features let Flownex model multi-module systems in which power, thermal, and fluid systems interact, suiting not only SMRs but also integrated energy systems and Power-to-X analyses. Importantly, Flownex provides a Python interface through which an external program can launch Flownex, open a project, establish the steady state, and then advance the transient one step at a time while reading outputs from or changing inputs of chosen components. This study builds the coupling on that interface, implementing for Flownex the three requirements identified from the earlier MARS-KS and MELCOR couplings: real-time variable exchange, external delivery of trip and actuation signals, and step-by-step time synchronization without drift.

3.2. Coupling Structure

In the coupling, Flownex occupies DICE's Physics-module slot and is solely responsible for the system thermal-hydraulics, while DICE advances the accident scenario through its Diagnosis, Reliability, and Scheduler modules. The two are connected by a coupling layer written in Python, which constitutes DICE's Physics module: it drives Flownex and passes its output to DICE. The calculation for one time step proceeds in three stages, read, evaluate, and apply (Figure 1). At the start of the analysis the coupling layer launches Flownex, opens the project, and establishes the steady state, and thereafter repeats these three stages at every step.

In the read stage, Flownex advances the transient by one step and the coupling layer reads the predefined variables and passes them to DICE. In the evaluate stage, DICE uses these values to check the actuation and recovery conditions of the safety functions, decides whether to split the current scenario into branches, and creates the corresponding states for each branch. In the apply stage, the coupling layer

converts DICE's decisions, such as reactor trip, valve opening and closing, pump trip, and system isolation, into input changes for the Flownex model and advances to the next step.

The two codes are synchronized by advancing them alternately, one step at a time: because Flownex hands over the updated state only after completing each step, the two time axes always coincide. Since all exchanged information passes through the predefined variable list and Flownex's standard features, any other thermal-hydraulic code that exposes its variables in the same way can reuse the identical scheme by changing only this intermediary layer.

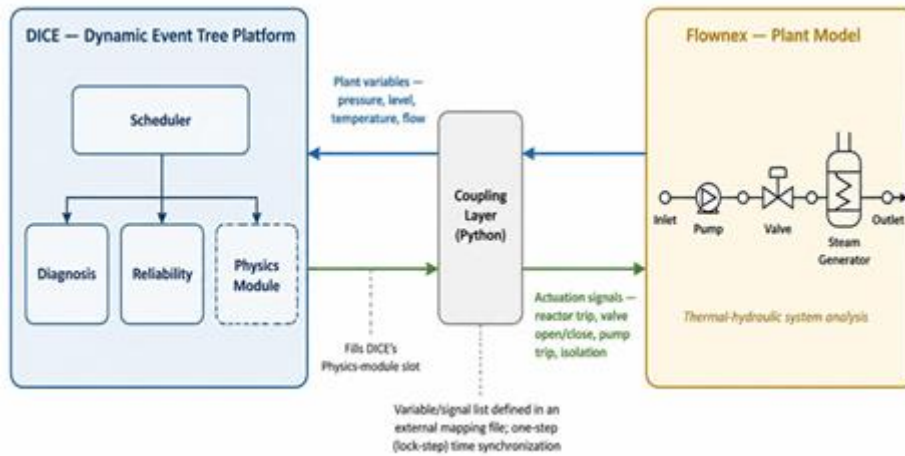


Figure 1. DICE–Flownex coupling structure.

3.3. Data Exchange

The information passed from Flownex to DICE consists of variables representing the thermal-hydraulic state, listed in a separate configuration file. For each variable the list records its name and type, which value of which component to read, and any unit conversion. When the analysis target changes, only this list is modified, so the number and type of exchanged variables can be changed freely. Because DICE judges actuation conditions and success criteria by variable name alone, differing naming conventions cause no problem.

The information passed from DICE to Flownex consists of safety-function actuation signals organized by actuation condition. One condition is linked to several input changes in the Flownex model; a single reactor-trip signal, for example, simultaneously inserts the control rods, trips the coolant pump, and closes the main steam isolation and feedwater valves. Each actuation can occur instantaneously or ramp over a set time, and once issued it is maintained at every step, so that a tripped pump does not restart on its own at the next step.

Each branch that splits off during the accident carries its own state: whether each safety function is undemanded, actuated, recovered, or failed; which component failures the reliability evaluation assumes; and the input to be delivered to Flownex. This per-branch state is stored as a single bundle and copied as a whole when moving to another branch, so changes in one branch do not bleed into another.

3.4. Coupling Procedure and Branch Restart

The procedure above is implemented as the coupling layer, a single Python program. At the start it reads once the information that does not change during the analysis: the variable list, the automatic and manual actuation rules, the success criteria and failure combinations, and the lists of actuation conditions, control variables, and failure modes. Thereafter, at every time step it repeats four stages in

a fixed order, namely state collection, rule evaluation, branch generation with probability calculation, and state output (Figure 2).

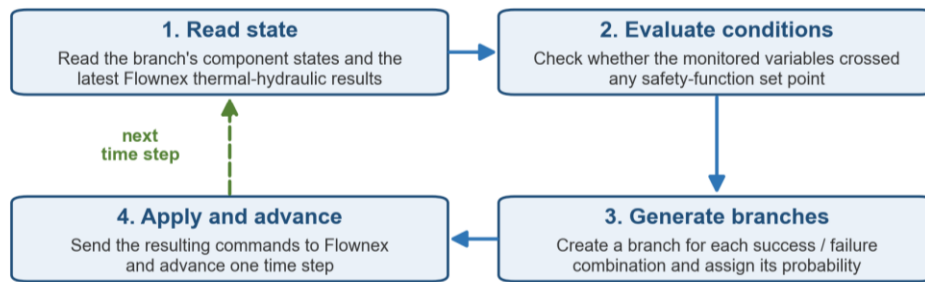


Figure 2. The four stages performed within one time step of the DICE–Flownex coupling: read state, evaluate conditions, generate branches, and apply and advance.

The probability of each failure mode is computed by the Reliability module from a per-component failure rate, distinguishing whether a mode is treated as a fixed per-demand probability or as one accumulated over the mission time; common-cause failures are also included. Because the branching logic works solely through the predefined variable list and the per-branch state bundle, coupling a different thermal-hydraulic code requires only a new connector for that code, while DICE's branching and reliability logic stay unchanged. This ease of adding modules or revising the reliability scheme is what makes the platform extensible.

The dynamic event tree is built in two phases. First, a single branch is run to the end while, at each set point, all branches that could split off according to the safety-function success criteria are identified and collected; these branches are then executed in turn. Because recomputing each branch from the start would be costly, the Flownex system state at the moment a set point is reached is saved and the branch is later continued from that point: when a saved state is restored, the previously maintained actuation commands are reinstated, the branch-specific actuations are applied, and the analysis continues. The deeper the tree, the more this saves computation; where state saving is unavailable, the scheme reverts automatically to recomputing from the start.

4. DEMONSTRATION: SMR LOCA SCENARIO

4.1. Hypothetical SMR Model in Flownex

The demonstration model is a hypothetical integral pressurized-water SMR built in Flownex with reference to publicly available integral-SMR design concepts [5]. On a component basis, the model represents the core with neutron kinetics, the steam generator, the primary and secondary coolant systems, and the containment that receives the discharged mass and energy.

The engineered safety features include reactor trip (RT); a four-train passive auxiliary feedwater system (PAFS), each train interlocked with closure of its MSIV; two pilot-operated safety relief valves (POSRV) protecting against reactor-vessel overpressure; an emergency depressurization valve (EDV) that discharges steam to depressurize the primary system; and an emergency recirculation valve (ERV) that returns steam condensed in the containment to the reactor vessel to establish natural circulation. Because the focus is the application of the coupling rather than the model itself, the model is described only as far as the accident scenario requires. The analysis begins after a 100 s steady state with a 0.5 s time step, and core damage is judged by the peak cladding temperature (PCT).

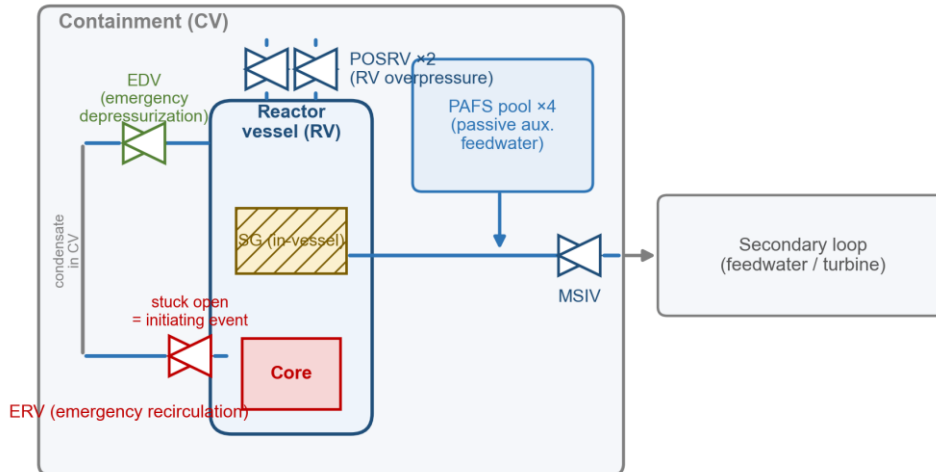


Figure 3. Schematic of the hypothetical integral-PWR SMR model used in Flownex.

4.2. Accident Scenario Definition

The initiating event is a LOCA in which one emergency recirculation valve (ERV1) sticks fully open over the first 30 s of the accident. Through the stuck-open valve, primary coolant is continuously discharged to the containment, raising the containment pressure and changing the primary-system pressure and level. As a result, when the actuation set points such as rising containment pressure, low pressure, or low level are satisfied, reactor trip and PAFS actuation are demanded, and as the accident progresses POSRV, EDV, and ERV may be demanded in turn.

At every time step, DICE's Diagnosis module checks whether monitored variables such as primary-system pressure, core power, containment pressure, and core water level reach each safety function's set point. When the primary-system pressure or core power leaves its set range or the containment pressure rises, reactor trip and PAFS are demanded; when the core water level falls below its set point, the EDV is demanded; and when the primary-system pressure exceeds the overpressure set point, the POSRV is demanded. At each demand the Reliability module generates the success, partial-success, and failure branches according to the KooN criteria and quantifies each branch probability with cut sets, truncating branches whose cumulative probability falls below a user-specified cutoff.

In principle the end state of each branch is classified by peak cladding temperature: a branch is safe if the PCT stays below the core-damage criterion (1204 °C) over the mission time and is core damage if it reaches that value. A grace period after each branch starts and a minimum interval between branches prevent spurious branches when steady-state values happen to satisfy a set point.

4.3. Results and Discussion

The coupling was applied to the ERV stuck-open LOCA of the hypothetical SMR to develop a dynamic event tree. From the initiating event, reactor trip, PAFS, and POSRV, together with EDV and ERV when required, were demanded over time according to the set-point rules, and branches were generated at each demand instant according to the KooN criteria. The accident sequences were thus enumerated automatically from the simulated system behavior rather than being specified in advance, which is the capability this demonstration is intended to show.

Figure 4 compares two representative branches. In the mitigated branch all safety functions operate: the reactor trips on its set point and the interlocked PAFS removes residual heat, keeping the primary-system pressure stable and the PCT low. Where PAFS does not secure residual-heat removal, the core water level falls to its set point, the EDV and ERV are demanded for emergency depressurization, and the primary system settles at a lower pressure. In both branches the PCT stayed well below the damage

criterion, showing that Flownex reproduces the behavior difference between safety-function success and failure and that DICE branches accordingly.

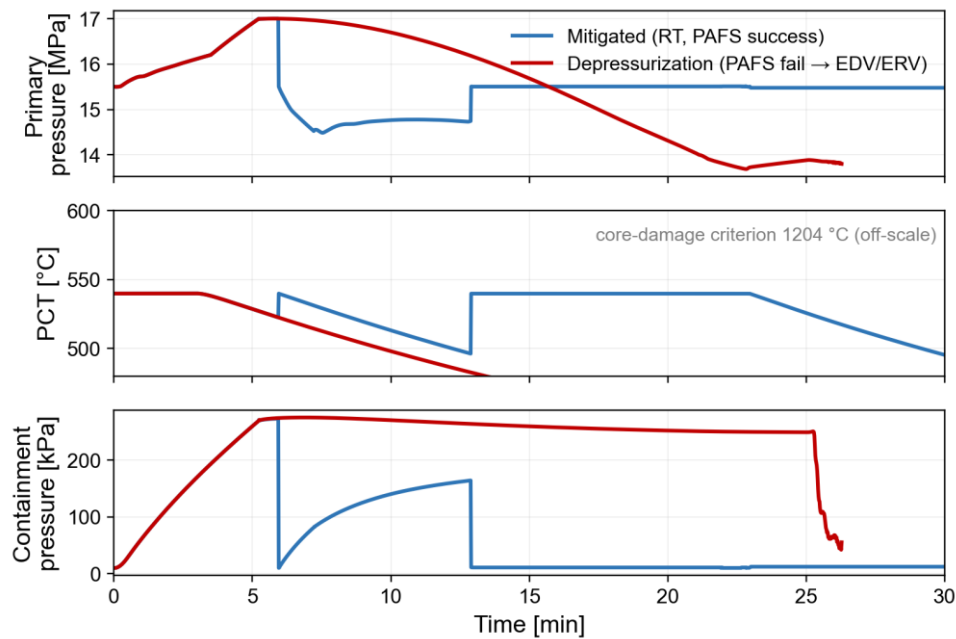


Figure 4. Thermal-hydraulic time histories for the mitigated and emergency-depressurization branches of the ERV stuck-open LOCA.

Figure 5 shows the structure of the event tree developed from the ERV LOCA. At each set point, success, partial-success, and failure branches are generated under the KooN criteria, and even for the same initiating event the functions later demanded and the branch timing differ with the system state at that moment. Along paths where reactor trip failed, the core power was not suppressed; in the present demonstration model such paths could not be advanced stably and terminated before mission completion. Branches that ran to completion kept the PCT well below the damage criterion.

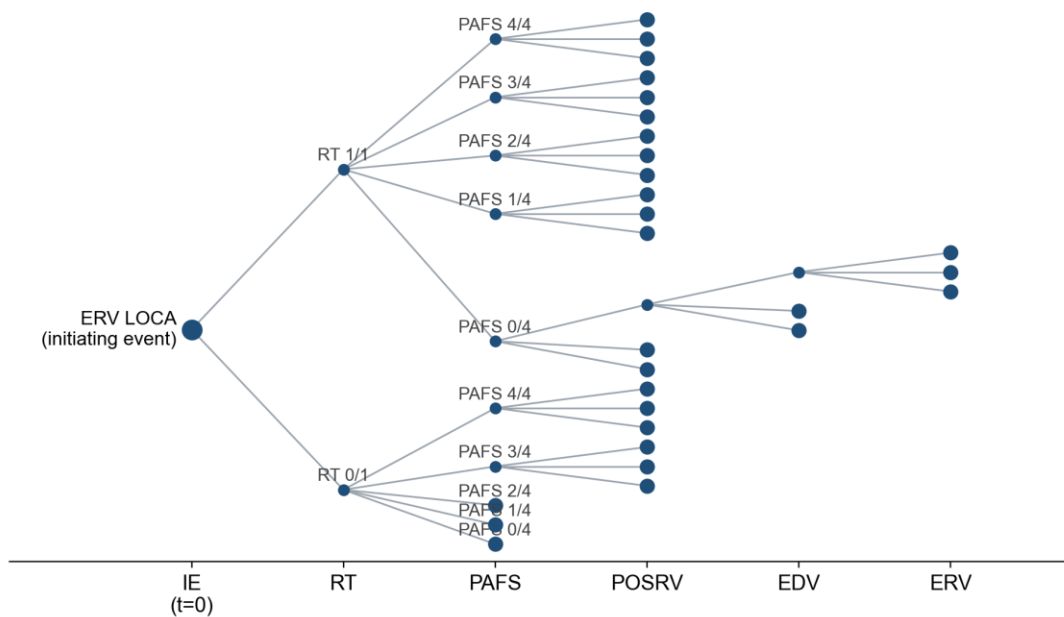


Figure 5. Dynamic event-tree structure for the ERV stuck-open LOCA.

These results show concretely where the dynamic event tree differs from a static one and why that matters for an SMR. In a static event tree the analyst fixes the order and timing of the safety-function demands in advance; in the DICE–Flownex coupling these are determined from the thermal-hydraulic state computed by Flownex, so the branch timing and the combination of demanded functions emerge on their own. The analysis therefore surfaces sequences that depend on timing and on partial success, such as the path on which PAFS fails to secure residual-heat removal and the falling water level then demands emergency depressurization by the EDV and ERV. For a reactor whose accident sequences are not yet catalogued, this automatic enumeration is the central practical benefit of the method.

Overall, the demonstration confirms that the DICE–Flownex coupling operates as intended, with Flownex advancing the system thermal-hydraulics and DICE generating and quantifying branches from that state. The branch probabilities reported here are illustrative, showing that the procedure performs per-branch quantification; refining the reliability data and improving model fidelity remain future work.

5. CONCLUSION

In this study, a general interface coupling the thermal-hydraulic system analysis code Flownex to the dynamic event tree platform DICE was designed and implemented. The data exchange is defined through a mapping file, giving a single-step, lock-step interface that is independent of the two codes' internals, and a Python coupling layer generalizes it so the thermal-hydraulic code can be replaced without changing DICE's branching or reliability logic; a snapshot-restart path at DDET set points was also provided. Applied to the ERV stuck-open LOCA of a hypothetical SMR, the coupling let the DET enumerate accident sequences automatically from the simulated behavior, rather than from a sequence list fixed in advance.

DICE can thus take a component-based system analysis code as its Physics module, in addition to its existing MARS-KS and MELCOR couplings, broadening its range of application. For SMRs in particular, whose accident sequences are still being established, the coupling supplies the realistic system-level physics that automated sequence discovery needs. Future work includes extension to externally induced sequences such as earthquakes, dynamic safety assessment of multi-module and multi-energy systems such as Power-to-X and integrated energy systems, and coupling with additional system codes.

Acknowledgements

This research was supported by the Nuclear Safety Research Program through the Regulatory Research Management Agency for SMRS (RMAS) and the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (No. 1500-1501-409) and National Research Council of Science & Technology (NST) grant by the Korea government (MSIT) (No. GTL24031-000)

References

- [1] U.S. Nuclear Regulatory Commission, “Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants,” WASH-1400 (NUREG-75/014), (1975).
- [2] T. Aldemir, “A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants,” *Annals of Nuclear Energy*, vol. 52, pp. 113–124, (2013).
- [3] G. Heo and D. Kwon, “Korean perspective for integrated deterministic-probabilistic safety assessment and its synergetic strategy with conventional methods,” *Nuclear Engineering and Technology*, vol. 56, no. 10, pp. 4031–4041, (2024).
- [4] S. Baek and G. Heo, “Development of dynamic integrated consequence evaluation (DICE) for dynamic event tree approaches: Numerical validation for a loss of coolant accident,” *Reliability Engineering & System Safety*, vol. 238, 109425, (2023).
- [5] K. K. Kim, W. Lee, S. Choi, H. R. Kim, and J. Ha, “SMART: The First Licensed Advanced Integral Reactor,” *Journal of Energy and Power Engineering*, vol. 8, no. 1, (2014).