Numerical Validation of Dynamic Event Tree Analysis Platform DICE^{TM} Physical Module by Preliminary Calculation

Hyunjoon Jeong [a], Jaeseung Suh [b] Gyunyoung Heo [c], Jonghyun Kim [d], Taewan Kim [a *]
Department of Safety Engineering, Incheon National University, 119 Academy-ro, Yeonsu-gu, Incheon, 22012, [a]
System Engineering & Technology, Room 302, 105 Sinildong-ro, Daedeok-gu, Daejeon, 34324 [b]
Department of Nuclear Engineering, Kyunghee University, 1732 Deokyoung-daero, Giheung-gu, Yongin, 17104, [c]
Department of Nuclear Engineering, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju, 61452, [d]

* *Corresponding author:* taewan.kim@inu.ac.kr

1. Introduction

The deterministic safety analysis (DSA) and conventional probabilistic safety assessment (PSA) methodologies have limitations in considering the time-dependent interactions of system process, equipment performance, and operator actions [1]. As an alternative methodology to avoid such limitations, it has been developed integrated safety assessment methodologies combining deterministic and probabilistic approaches to take into account of dynamic interactions during transients [2,3]. Meanwhile, the integrated dynamic probabilistic safety assessment (IDPSA) tool namely DICE (Dynamic Integrated Consequence Evaluation) has been developed on the basis of dynamic event tree (DET) methodology, aiming to provide a more comprehensive evaluation for the IDPSA [4-7]. In this paper, specific features of the DICE will be introduced and the numerical validation of DICE-MELCOR is discussed using the preliminary calculation for a simple model. Also, the study aims to provide straightforward guidelines for converting system code input to the DICE physical module input, focusing on the MELCOR code.

2. Structure and process of DICE

2.1 Structure of DICE

DICE has been developed as a prototype to demonstrate the essential algorithm and key concepts, as shown in Figure 1. The scheduler of DICE is responsible for the overall interaction between each module. Furthermore, the DICE comprises a physical module that simulates the thermal-hydraulic behavior of the system, automatic and manual diagnostic modules to determine whether a branching condition has been reached based on real-time thermal-hydraulic analysis results, and reliability module that reflects the performance of safety systems.

2.1.1 Scheduler

The scheduler is one of the core component of DICE which manages information exchange between modules. It generates and decomposes event sequences based on interaction information, deciding branching in event trees (ETs). The simulation's continuation or termination depends on the plant and physical model status controlled by the scheduler. The physical model stops simulation according to the signal generated by the scheduler if predefined conditions are met. Additionally, the scheduler may halt calculations if the probability of an event sequence is very low, categorized as 'Cut-off.'

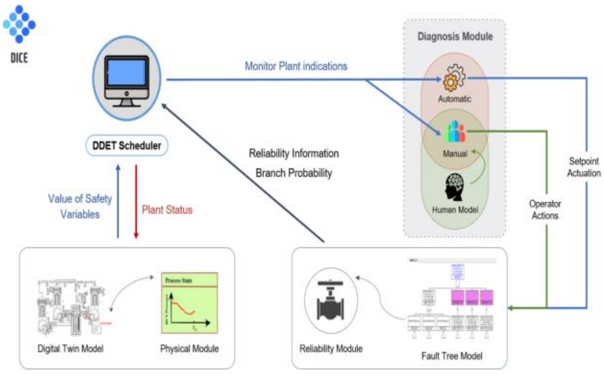


Fig. 1. Schematic diagram of DICE structure [8]

2.1.2 Physical module

The physical module in DICE uses a thermal-hydraulic safety analysis code to simulate the physical behavior of the system. The results of the safety variable calculation at each time step are transmitted to the scheduler, while the scheduler provides input settings for branch conditions. The physical module acts as a connection between various system codes and the scheduler, allowing flexibility in connecting with different simulation codes. DICE is designed to connect with various simulation codes, extending its capability to any system analysis codes with consistent protocol maintenance. Currently, DICE focuses on analyzing internal events for Level 1 PSA and Level 2 PSA, embedding MARS-KS [9] and MELCOR [10], respectively, in the physical modules.

2.1.3 Diagnosis module

The diagnosis module in DICE is crucial for establishing branching rules and assessing the operational status of the plant components and systems. These rules, formulated with conditional expressions, compare received plant variables with predefined setpoints and logical expressions aligning with safety system operating conditions. When branches are initiated automatically or manually, the diagnosis module makes controls by evaluating defined criteria and logic expressions. Branching initiations use branching rules in logical expressions, but there is a difference in presenting results. For automatic operation, a single branching point generates branches based on the successful combinations of the systems, as shown in Figure 2. In contrast, for manual operation, multiple branching points generate branches based on the timing of actions, resulting in a division over time.

In summary, the diagnostic module receives safety variable information from the scheduler during simulation and diagnoses branching rule fulfillment. Then, it reallocates branch operating conditions in the physical module, returning them to the scheduler if specific rules are satisfied.

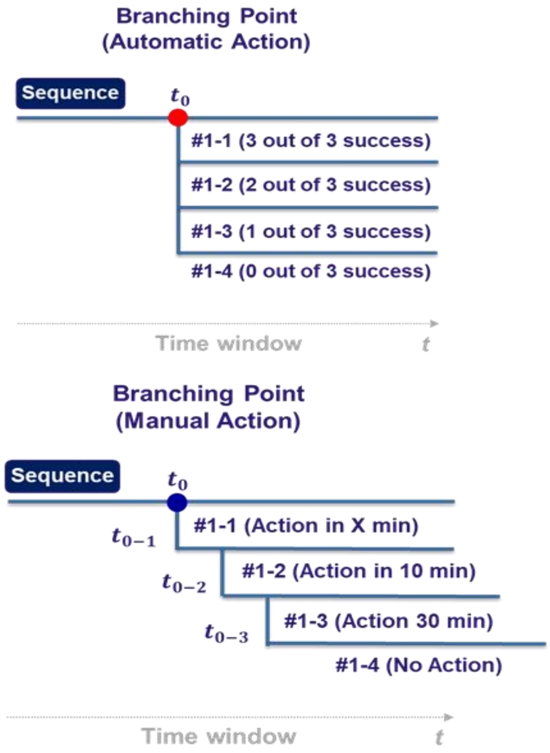


Fig. 2. Branching form of DICE according to diagnostic module action type

2.1.4 Reliability module

Upon the diagnosis module determination of a branching occurrence, the reliability module transmits the generated branch information to the scheduler, considering the failure type of the nuclear power plant (NPP) systems and equipment in each branch. Reliability data for each system is derived from Fault Trees (FTs), and a distinct computational engine quantifies the probability of branch and event sequence. For a more comprehensive understanding of the DICE reliability module, detailed explanations can be found in the reference paper [11].

2.2 Progress of DICE

At the onset of the simulation, safety variable values calculated by the physical module at each time step are transmitted via the scheduler to the diagnostic module. The diagnostic module assesses whether the received safety variables meet predefined conditions, determining whether to proceed to the next time step or initiate branching at that moment.

Upon confirming branching, the diagnostic module determines the number of branches generated and provides control information for each branch to the physical module. Subsequently, the reliability module quantifies each generated branch, calculating branching probabilities. Finally, the scheduler receives information including the number, form, control details, and probabilities of the branches created at that branching point. Utilizing this information, the scheduler constructs the Discrete Dynamic Event Tree (DDET) and continues with the simulation. The form of the DDET created during the actual simulation based on this mechanism is illustrated in Figure 3.

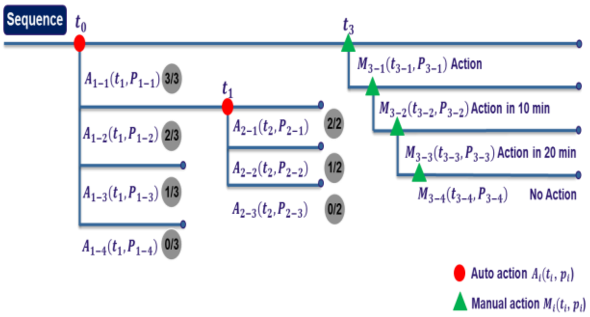


Fig. 3. DDET schematic diagram through branching of DICE

3. Interface between DICE and system code

3.1 Numerical consistency of DICE physical module

The most crucial in constructing the physical module of DICE should be that the thermal-hydraulic code used in the physical module produces identical results to the standalone version during both information exchange with external code and modifications for reflection. To reflect decisions from other modules into the physical module, modifications are needed to update real-time controllable variable values during the analysis. This process introduces the possibility of different results compared to the original analysis. However, the inclusion of the thermal-hydraulic code in the physical module should not lead to any modifications that could impact the results of the analysis.

MARS-KS provides the variable update function using interactive variables, and a previous study has confirmed the numerical validation of DICE-MARS [12]. The MELCOR code provides a connection method with other codes through the Analytical Control Function (ACF) method since Version 2.2.X.

3.2 Method for building physical module input

The standalone calculation of the system code automatically reflects the operational status of equipment when the thermal-hydraulic variables reach set values, progressing with the calculation. Such calculations can be used for simulating a single case. However, DICE, as it needs to analyze multiple cases simultaneously, has the characteristic of controlling various physical models with different operational states under different conditions. Therefore, the DICE physical module continuously receives the diagnosis results from the diagnostic module after the initial simulation has started.

To reflect these characteristics, variable exchange between the physical module and the scheduler is crucial. The variables exchanged from the physical module could be broadly categorized into monitoring variables that can monitor the status of NPPs, such as temperature, pressure, and flow, and control variables that can control the operational status of equipment. Monitoring variables are used to support the diagnostic module in determining branch conditions, while control variables are used to assign the device's operational status for each branch. Therefore, the conversion is required from the system code input to the DICE physical module input to analyze multiple cases simultaneously. In order to assist the conversion, the following five steps has been developed, as illustrated in Figure 4.

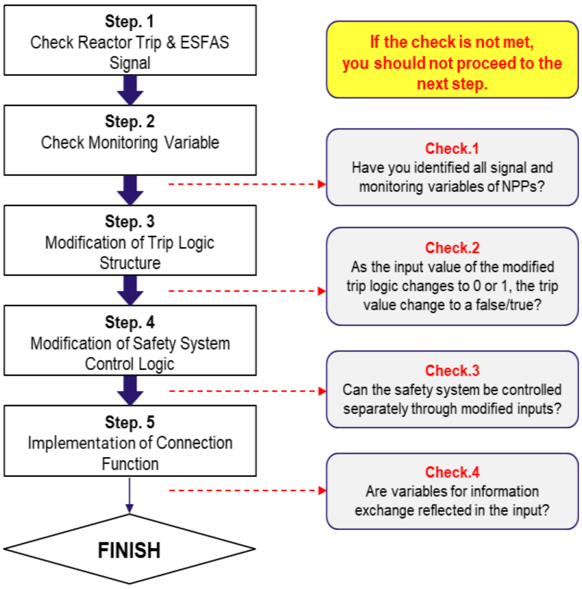


Fig. 4. Flowchart of DICE physics module input transformation procedure

3.3 Description of flowchart of DICE physics module input transformation procedure

In this section, the five steps of the DICE physical module input conversion method is discussed using examples of MELCOR.

The first step is to verify the signals that activate the reactor protection system (RPS) and engineered safety features (ESFs) in the NPP. This is primarily done to identify which trip signals are connected to the safety system, initiating a change in equipment status. Also, it is a preliminary confirmation to later ensure the correct implementation of trip logic structure changes in Step 3.

The second step is to identify the cause of the signals identified in the previous step. For example, if a low pressurizer pressure (LPP) signal leads to the generation of a safety injection system signal, the monitoring variable at this time becomes the pressurizer pressure. Representative monitoring variables include pressure, temperature in the primary and secondary systems, and collapsed water level of the system, flow rate of the system, [1]peak cladding temperature (PCT).

When the monitoring variable reaches a specific set value during transient calculations, the signals for the operation of the RPS and ESF are triggered, causing a change in the operational status of equipment. However, to reflect the simulation features of DICE, the trip logic, which implements the automatic operation of devices, needs to be modified to allow manual control based on the information received from the scheduler. In other words, when the monitoring variable reaches the set value, the scheduler needs to change the value of the corresponding control variable to change the corresponding trip status. This process is required in step 3 and done by using the ACF function in case of MELCOR.

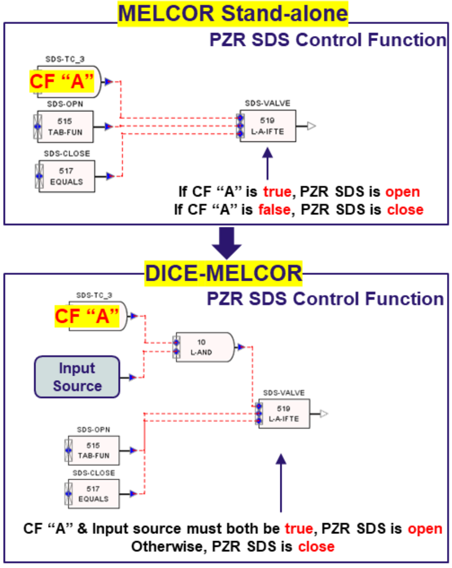


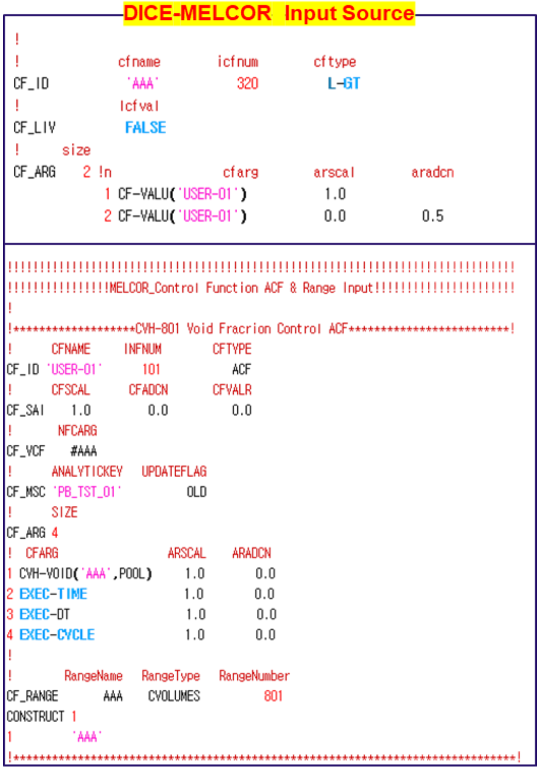
Fig. 5. Example of DICE-MELCOR input implementation

---

[1] This is not a general monitoring variable, but a variable to terminate calculation when the limit value is exceeded.

Step 4 involves incorporating the previously modified trip logic into the equipment system. In the case of the MELCOR standalone, when the thermal-hydraulic variables satisfy specific conditions and the value of a control function, 'A,' becomes true, the equipment connected to 'A' will operate. However, in the DICE physical module, the equipment should only operate when both the value of the control function 'A' and the value of the ACF function (input source) are true, as shown in Figure 5.

Step 5 is the configuration of the input source. As mentioned, the MELCOR code could implement connection with the external system through the ACF function. Therefore, the input source for external integration is controlled by the ACF function, and such input takes the form shown in Figure 6. The CF_ARG value of the input source is defined as the ACF function called USER-01, which is a function controllable from the external system through the DLL interface. Consequently, through this interface, the connection of variables and the resulting reflection in the equipment state could be simulated.



DICE-MELCOR Input Source

```
!
!             cfname        icfnum        cftype
CF_ID          'AAA'           320          L-GT
!             lcfval
CF_LIV         FALSE
!       size
CF_ARG    2 !n                  cfarg        arscal         aradcn
              1 CF-VALU('USER-01')            1.0
              2 CF-VALU('USER-01')            0.0             0.5
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!MELCOR_Control Function ACF & Range Input!!!!!!!!!!!!!!!!!!!!!!!
!
!*****************CVH-801 Void Fracrion Control ACF*************************!
!      CFNAME      INFNUM        CFTYPE
CF_ID 'USER-01'      101            ACF
!      CFSCAL       CFADCN        CFVALR
CF_SAI    1.0          0.0           0.0
!       NFCARG
CF_VCF    #AAA
!      ANALYTICKEY   UPDATEFLAG
CF_MSC 'PB_TST_01'          OLD
!      SIZE
CF_ARG 4
!  CFARG                  ARSCAL    ARADCN
1 CVH-VOID('AAA',POOL)      1.0        0.0
2 EXEC-TIME                 1.0        0.0
3 EXEC-DT                   1.0        0.0
4 EXEC-CYCLE                1.0        0.0
!
!         RangeName   RangeType   RangeNumber
CF_RANGE        AAA    CVOLUMES          801
CONSTRUCT 1
1          'AAA'
!*************************************************************************!
```

Fig. 6. Example of DICE-MELCOR input source and ACF input

4. Numerical validation of DICE-MELCOR code

4.1 Description of simple model

In this study, a simple model was developed to verify the numerical accuracy and consistency between the two codes, as shown in Figure 7. The simple model was designed with a simplified structure to evaluate and validate the basic functionality of MELCOR for the clarity of results analysis. The simple model consists of a total of 7 control volumes (CV) and 3 flow paths (FP), with the implementation of control functions to control the pressure of CV300, the opening ratio of FP500, and the flow rate of FP600. The 0/1 signals of both codes are implemented as void fractions of arbitrary control volumes CV801, CV802, and CV803. The aim is to evaluate the fundamental capabilities of the physical module using MELCOR and to simulate the changes in the plant state.

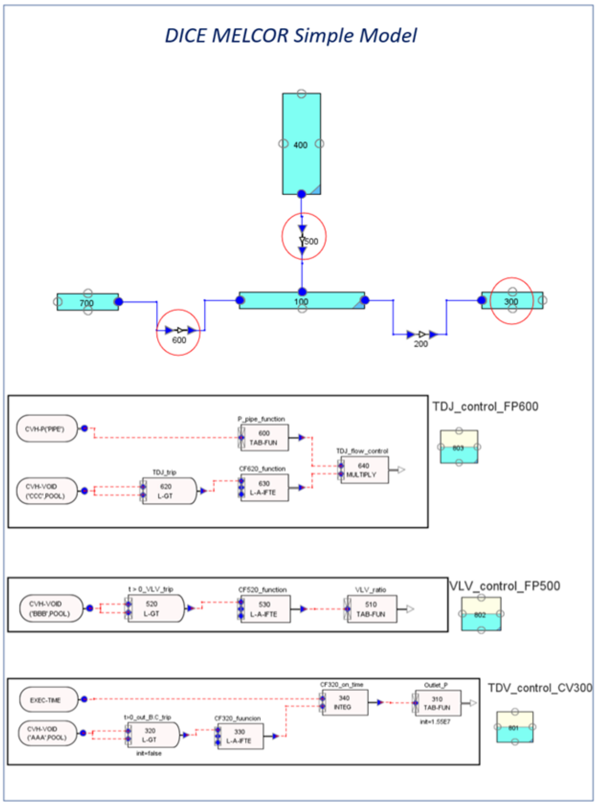


Fig. 7. Simple model for DICE-MELCOR preliminary calculation

4.2 Result of preliminary calculation

Figure 8 illustrates the CV801-803 void fraction for the physical module of the DICE and the MELCOR stand-alone. It shows that the void fraction value changes to 1 after 100 seconds. The values of the MELCOR stand-alone are controlled using a table function and, in case of the DICE physical module, those are controlled by the ACF connections. To verify numerical accuracy and consistency between the two codes, the thermal-hydraulic variables controlled by the aforementioned methods were compared. Among them, Figure 9 presents the pressure of the CV300. As shown in Figure 9, a change in pressure is observed from the point where the void fraction value becomes 1. It means that the new interface connection through the ACF was performed without problems. Through this, it is judged that it will be possible to reflect changes in the status of the power plant according to the status of the equipment and operator actions.

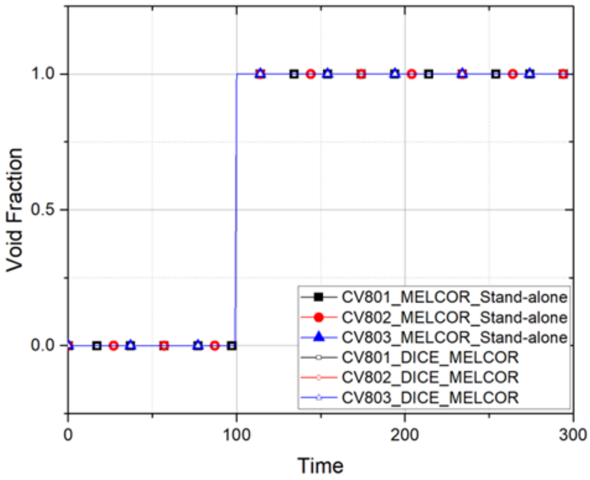


Fig. 8. Void fraction of control volumes

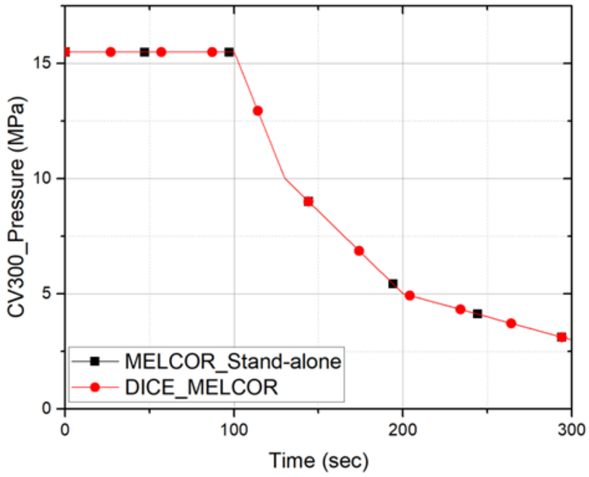


Fig. 9. Pressure of control volume No.300

5. Conclusions

In this paper, specific features and the overall progress of the DICE were introduced. Additionally, the study provides straightforward guidelines for converting the MELCOR input to the DICE physical module input for the numerical validation of the DICE-MELCOR. The preliminary calculation was performed using the physical module of the DICE and the MELCOR to demonstrate the integrity of the DICE physical module. Both codes yield identical results, indicating that the two codes possess the same interpretative capability from the initial initialization step through the iterative calculation process to the calculation end step. Therefore, it can be concluded that there is no difference from the existing the MELCOR stand-alone code calculation result, affirming the integrity of the MELCOR coupled in the DICE.

REFERENCES

[1] GE. Wilson, Historical insight in the development of Best Estimate Plus Uncertainty safety analysis, Annals of Nuclear Energy, 52, 2-9, 2013.
[2] T. Aldemir, A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants, Annals of Nuclear Energy, 52, 113-124, 2013.
[3] Durga Rao K, Chapter X Dynamic PSA. Reliability and Safety Engineering. 1996.
[4] S. Lee, S. Baek, G. Heo, T. Kim, J. Kim, Development of DICE (Dynamic Integrated Consequence Evaluation) for procedure coverability studies: conceptual design phase, Transactions of the Korean Nuclear Society Autumn Meeting, 2018.
[5] S. Baek, G. Heo, T. Kim, J. Kim, Introduction to DICE (Dynamic Integrated Consequence Evaluation) toolbox for checking coverability of operational procedures in NPPs, The Annual European Safety and Reliability Conference ESREL, 2019.
[6] S. Baek, T. Kim, G. Heo, J. Kim, Branching rules and quantification in dynamic probabilistic safety assessment: development of DICE (Dynamic Integrated Consequence Evaluation), Transactions of the Korean Nuclear Society Autumn Meeting, 2019.
[7] S. Baek, T. Kim, J. Kim, G. Heo, Application of DICE (Dynamic Integrated Consequence Evaluation) case study on branching rules examples, Transactions of the Korean Nuclear Society Virtual Spring Meeting, 2020. July 9-10.
[8] S. Baek, 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, 2023
[9] Korea Institute of Nuclear Safety (KINS), MARS-KS Code Manual Volume I: Theory Manual, KINS/RR-1882 Vol.1, 2018.
[10] L.L. Humphries et al., MELCOR Computer Code Manuals: Version 2.2.19018. SAND2021-0252 O, Sandia National Laboratory. (2021)
[11] G, Heo., S, Baek., D, Kwon., H, Kim., J, Park., “Recent Research towards Integrated Deterministic-Probabilistic Safety Assessment in Korea.” Nuclear Engineering and Technology (2021).
[12] H. Jeong, J Kim, G. Heo, T. Kim, Verification of DICE physical module integrity by SBLOCA calculation, Transactions of the Korean nuclear society spring virtual meeting online; 2020.