Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 18-19, 2023

Prototype Development of DICE-PCTRAN for Dynamic Scenario Analysis

Yongjoon Lee, Gyunyoung Heo*
Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Yongin, Republic of Korea
*Corresponding author: gheo@khu.ac.kr

1. Introduction

In the meantime, Dynamic Integrated Consequence Evaluation (DICE) has been developed as an analysis tool for Dynamic Event Tree (DET) analysis [1,2].

The need for analysis of context-driven changes by, for instance, such as operator intervention has been always highlighted not to lose any insights caused by over- or under-conservatism. However, as the scope of risk assessment in Nuclear Power Plants (NPPs) expands, the existing Probabilistic Safety Assessment (PSA) need to be improved for analyzing such complexity [3].

Currently, DICE consists of a scheduler that manages three modules: physical module, diagnostic module, reliability module. For the physical module, Multi-dimensional Analysis of Reactor Safety (MARS) and MELCOR are used.[1].

As another potential option, the PCTRAN software, which can be a physical module, has its own simplified user interface (UI) and can present an intuitive mimic to users in real time. Therefore, in this study, we attempted to introduce a prototype of DICE using PCTRAN (Physical module).

2. Background

2.1 Summary of DICE™

DICE supports the analysis of dynamic event tree. In a conventional PSA, it consists of a Fault Tree (FT) and an Event Tree (ET) to generate a set of branch and quantified results while contributing to improved safety. Figure 1 shows the schematic diagram of the DICE workflow. DICE uses Discrete Dynamic Event Tree (DDET) or Monte Carlo Discrete Event Tree (MCDET) rather than a traditional ET. If the multi-branch applying the DDET method is satisfied with the branching rule, it is divided into multiple branches which are registered in advance. In the case of a single branch applying the MCDET method, the branch is maintained as one path. In order to drive both event tree generations, the DICE is composed of four components: scheduler, physical module, diagnostic module, and reliability module [1,2].

The scheduler is responsible for creating new branch points and checking for branch of potential accident scenarios. In addition, it is responsible for exchanging and sharing information between models at specified time intervals and under specific conditions, assigns boundary conditions for thermal hydraulic analysis, and sets operation shutdowns. The physical module refers to the dynamic interaction of each component with the thermal hydraulic system of NPP.

Currently DICE embeds physical modules including MARS and MELCOR, and PCTRAN would be introduced in this study. This model is based on plant design and operation information, heat and heat transfer, and physical characteristics of each facility. The reliability module predicts system failures and prevents or recovers from them. To this end, all parts and equipment of the power plant are monitored, and reliability is calculated to ensure safety. The diagnostic module plays a role in diagnosing problems in the power plant system, detects signs of failure, and analyzes the cause to solve the problem.

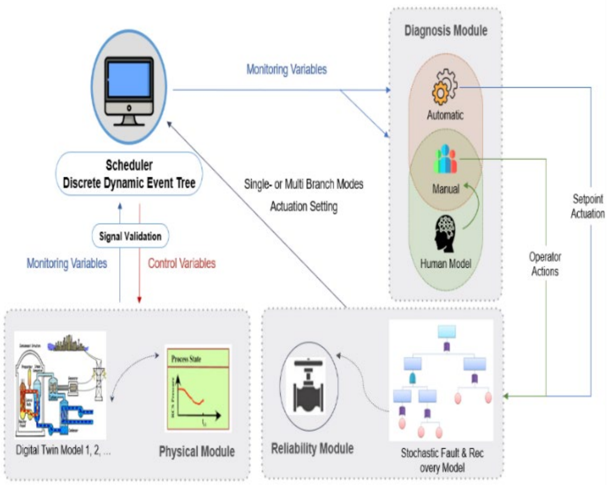

Fig. 1. Schematic diagram of DICE

2.2 Brief Description to PCTRAN

As shown in Figure 2, PCTRAN is an NPP simulation software that can be implemented on a personal computer. This software was developed by Micro-Simulation Technology (MST). PCTRAN provides a user-friendly graphic interface to directly manipulate graphic elements (pumps, valves, etc.) to operate the simulation software. PCTRAN is programmed based on an open source (simple formula), so it is not able to quantitatively express finer values than other physical modules, but it can run faster than the actual time. Also included is a severe accident model for NPPs, and these features allow PCTRAN to be used to evaluate accidents and their consequences. PCTRAN complies with Microsoft Windows 10 or higher environment and processes data output in Access database format of MS office [2].

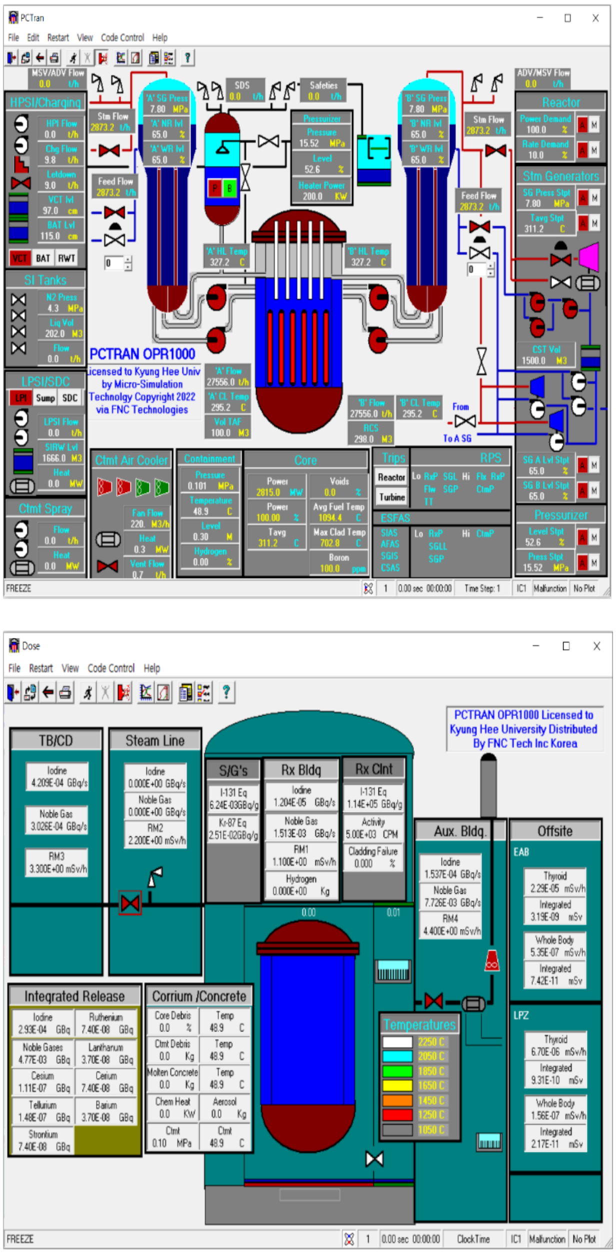

Fig. 2. Graphical user interface of PCTRAN
(Upper: Main Mimic, Lower: Dose Mimic)

3. Methodology

The basic structure of the DICE-PCTRAN prototype is shown in Figure 3. When the simulation starts, the physical variables of PCTRAN are passed to the diagnostic module Python. When the branching rule set in the Python algorithm is satisfied, the probability of the control condition (success criterion) of the physical module set in the Python algorithm is reflected. Afterwards, the python algorithm signals the macro, and the automatic mouse/keyboard works to activate the PCTRAN equipment.

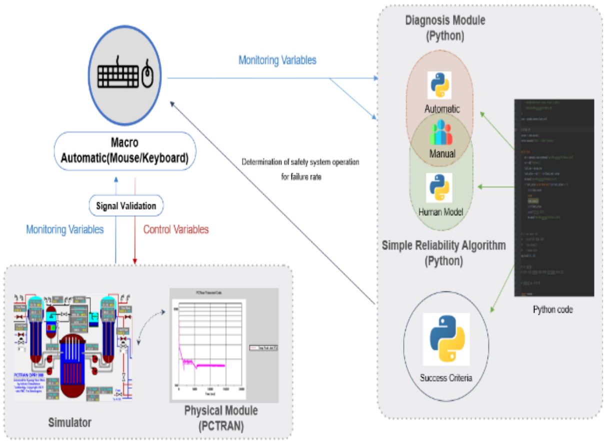

Fig. 3. Schematic diagram of DICE-PCTAN Prototype

In the method of connecting the existing physical module and the diagnostic module of DICE, the diagnostic module receives the variables sent by the physical module. The diagnostic module analyzes physical variables, interacts real-time variables between the physical module and the diagnostic module, and interprets previously undiscovered scenarios according to branching rules. However, PCTRAN needs to operate the power plant simulator with mouse/keyboard manipulation for equipment operation/initial condition setting. Therefore, as an idea on how to connect PCTRAN and DICE, a macro program that can manipulate the UI for PCTRAN equipment operation/initial condition setting is needed. As an idea to overcome these limitations, the diagnosis module was replaced by using Python, an external program that supports macro programs and branching rules at the same time.

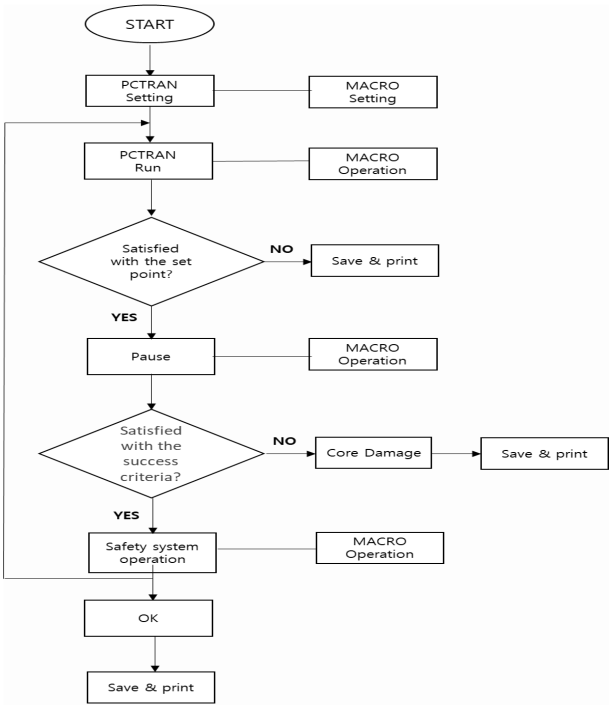

Fig. 4. Flowchart of PCTRAN & MACRO

Figure 4 is a flow chart of using Python- coded' macro program to replace the physical module with PCTRAN and the diagnostic module. Python macro program can be used to set the initial conditions of PCTRAN and execute the Run operation. If the branching rule is satisfied, the stop button is pressed using the macro, and if the success criterion is also satisfied, the graphic operation is performed using a macro according to ET. The plant state is checked according to the procedure and repeated until the final state is reached.

```
Input : accident_set, accident_type, severity simulation_time
macro do(input)
macro do(Run)

While true
        If variable satisfy set point
        Break
        macro do(pause)

If satisfy success criteria
        macro do(safety system)
        macro do(Run)

        While true
                If variable satisfy set point
                Break
                macro do(pause)

        If satisfy success criteria
                macro do(safety system)
                macro do(Run)
                        :
                wait next signal
                macro do(pause)
                print("OK")
                save OK Data

        else
                wait next signal
                macro do(pause)
                print("CD")
                save CD Data
else
        wait next signal
        macro do(pause)
        print("CD")
        save CD Data
```

Fig. 5. Macro Algorithm Implemented with Python

Figure 5 presents an algorithm that sets the initial conditions of PCTRAN using the macro program, a diagnostic module, and creates branches through interaction between physical variables and equipment/operator modules using the macro program.

The coordinates for setting the initial condition of PCTRAN and the coordinates of the equipment are preset in the Python algorithm. After waiting until the physical variable received from PCTRAN satisfies the branch condition, the next branch is created according to the branch condition and success criteria. The automatic mouse/keyboard operates according to the branch condition and success criterion, operates the PCTRAN equipment, and repeats this method until the end point is reached.

4. Results

In the results, a video Uniform Resource Locator (URL) of a condensed process is attached to help us understand the DICE-PCTRAN prototype [5].

The algorithm based on the Large Loss Of Coolant Accident (LLOCA) scenario operates the mouse/keyboard suitable for branch conditions and success criteria and transmits them to PCTRAN.

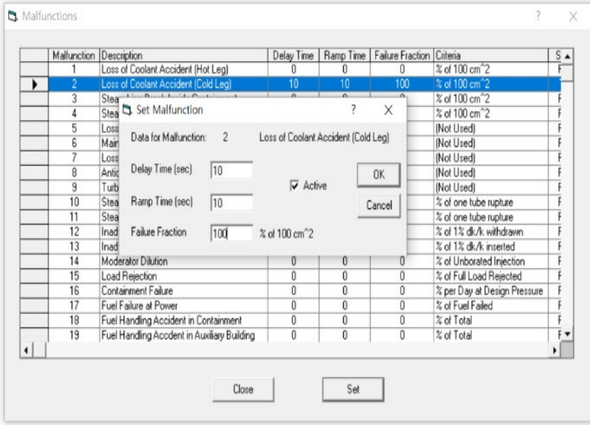

Fig. 6. Initial scenario setting using mouse/keyboard.

Figure 6 presents the UI for setting accident type, severity, and simulation time using automatic mouse/keyboard manipulation, an LOCA scenario is selected using Python macro input, delay time is 10 seconds, ramp time is set to 10 seconds, and the Failure Fraction is set to 100.

To avoid automatic injection of the safety system programmed in PCTRAN before starting, malfunction state is adopted as highlighted by the red box to operate the safety system corresponding to the ET of LLOCA as a signal from the macro program.

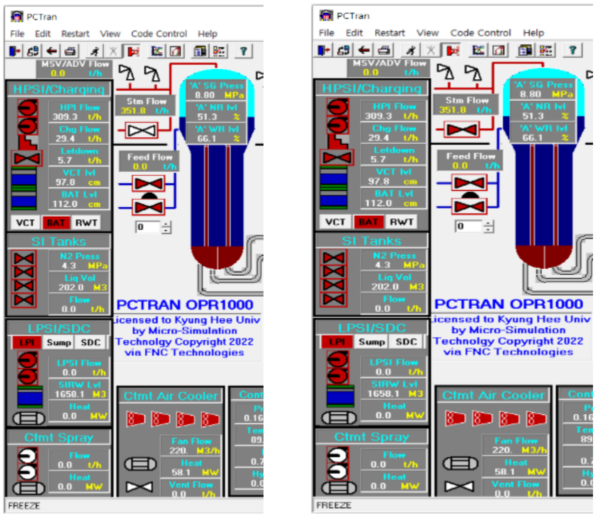

Fig. 7. Safe injection by manipulating the mouse.
(Left: HPH ON, Right: HPH OFF)

In Figure 7, the PCTRAN equipment operated the equipment (pump, valve) of the simulator with automatic mouse/keyboard operation through the macro program based on the branch setting conditions of the LLOCA scenario of DICE's diagnostic module. It shows whether the injection valve is opened/closed according to the success of the safety system corresponding to the ET of LLOCA

5. Conclusions

In this study, we proposed a method for developing a DICE-PCTRAN prototype using a macro program, taking the LLOCA event scenario as an example. This approach corresponding to the diagnostic module of DICE enable to provide a user interface and offer various advantages.

In particular, simultaneous operation of various branches using PCTRAN (physical module) is required for fast analysis, and completely separate models for diagnosis and reliability modules are required. Future work will introduce improved algorithms to fully separate the diagnostic and reliability modules in DICE-PCTRAN. It is expected that these algorithms enhance the accuracy and efficiency of prototypes, making them valuable tools for risk assessment and safety analysis.

Acknowledgments

This work was supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KOFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea. (No. 2103081)

REFERENCES

[1] Baek, S., Kim., T, Kim, J., Heo, G. “Branching Rules and Quantification in Dynamic Probabilistic Safety Assessment: Development of DICE (Dynamic Integrated Consequence Evaluation)”, Korean Nuclear Society Autumn Meeting, Ilsan, Korea, 24-25 October 2019.

[2] Heo, G., Baek, S., Kwon, D., Kim, H., Park, J. (2021). Recent research towards integrated deterministic-probabilistic safety assessment in Korea. Nuclear Engineering and Technology. 2021.

[3] Heo, G., Baek, S., Kwon, D., Gwak, Y. (2023). Korean Perspective for Integrated Deterministic and Probabilistic Safety Assessment and it Use. Proceeding of ICAPP .2023.

[4] Ben Qi1, Xingyu Xiao1, Jingang Liang. (2022). An open time-series simulated dataset covering various accidents for nuclear power plants Nature. Scientific data.2022.

[5] Lee, Y. 2023. “Prototype Development of DICE-PCTRAN for Dynamic Scenario Analysis”, March 18, 2023, Video, 4:08. https://youtu.be/9BfFKTVjw7Y.