Software Platform to Assimilate Simulation Modules with Referenced Plant Data

Sungchul Lee*, Minkook Kwon, Boyeon Lee, Keunbyul Lee
WHEELERS Inc., 1109-ho, 604, Joongbu-Daero, Yongtong-Gu, Suwon-Si, Korea
Corresponding author: sc.lee@wheelers.biz

Keywords : Module Based Platform, Bigdata, Artificial-Intelligence, Machine learning, Modeling and Simulation

1. Introduction

Data from nuclear power plants can be divided into two types in terms of machine learning or deep learning of artificial intelligence technology. In other words, it is 'data with driving experience' and 'data that is likely to be experienced' by the operator.

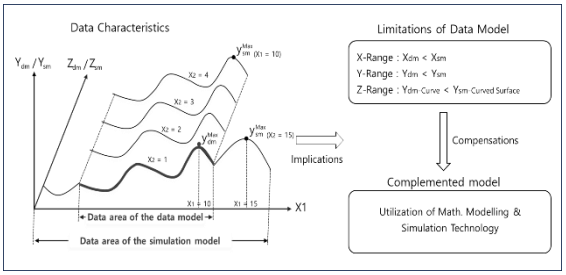

Fig. 1 Data model vs simulation model

Usually, data with experience operating in a power plant are stored as historical data in a power plant computer, and data that may be experienced are generated through simulation by mathematically modeling the system based on scientific knowledge. In terms of data quantity, the more precise the results are targeted, the larger the amount of simulation data as shown in Fig. 1. The importance of modeling and simulation with reasonable performance is absolute.

On the other hand, the engineering fields related to nuclear power plants related to simulation consist of neutron dynamics, thermodynamic mechanics, electrical engineering, control engineering, chemical engineering, and radiological engineering. Several to hundreds of different types of codes are used in each of these fields, and these codes are used by experts in each field to analyze them through simulations. An integrated environment that supports the familiar codes of experts in these fields continuously and generates big data through modeling and simulation is absolutely necessary to lead the global market with the advancement of the industry by providing user convenience functions and real-time call environments.

Therefore, in this study, we would like to describe the design and development of an integrated environment that helps each expert in the power plant use various codes and models conveniently.

2. Target Functions of The Software Platform

The target functions of this Research and Development platform are as follows.

First, it should be possible to accommodate a wide variety of codes. For example, a code currently used by someone should be implanted without any change in performance, and it should be easy to connect with codes in other fields. That is, the fluid simulation code should be able to easily link with the electric simulation code or the control simulation code driving it.

Second, the foundation of platform software should be an environment commonly used by users reflecting the latest technology.

Third, there should be no limitation on the implantation of a new model or the increase in the number of nodes through the finite element analysis method. This becomes the basis for installing a measurement control system including a sensor that is not currently found in nuclear power plants, or for extremely precise simulation of behaviors such as vibration and heat dissipation.

3. Architecture of The Software Platform

To achieve the purpose described in the previous section, the detailed content of MBStation (Module Based Station) developed by WHEELERS Corporation is as follows:

3.1 Architecture of MBStation

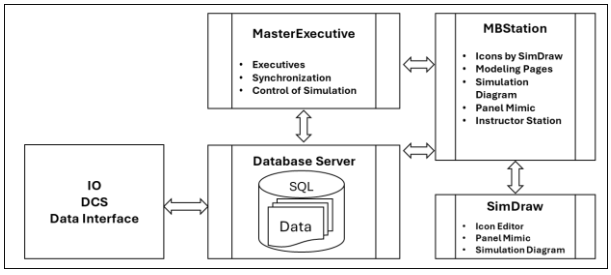

Fig. 2 Architecture of software architecture

The MBStation is the Module Based Station. It is to be developed as a software platform to assimilate simulation modules with referenced plant data (data from experience) to generate data to be experienced to facilitate machine learning by Artificial-Intelligence algorithms.

This platform includes:

- Database Server for simulation modeling data management;
- Master Executive for simulation model scheduling and synchronization;
- Module Based Station (MBStation) provides a graphical user interface that allows for editing, testing, and monitoring the development and performance of simulation models. It also serves as an Instructor Station for simulator operations. The station contains the following:

a. A two-dimensional(2D) graphics editor for editing graphical objects and providing object animation;
b. A basic module library set for system model development, including logical systems, control systems, single-phase flow network systems, and electrical systems.

- Users can use the wizard to build their own modeling objects;
- Interfaces of panel Input-Output(IO) or Distributed Control System(DCS);
- Steam table search tool.

3.1.1 Database server

The simulation model consists of hierarchical components, including modules, pages, nodes, and links between nodes. All of these components will be served by the Database Server for client requests, including Master Executive and MBStation. The Database Server uses the HTTP interface protocol to communicate with clients. Therefore, it is even possible to remotely access the services provided by the server through internet connections as shown in Fig. 3.

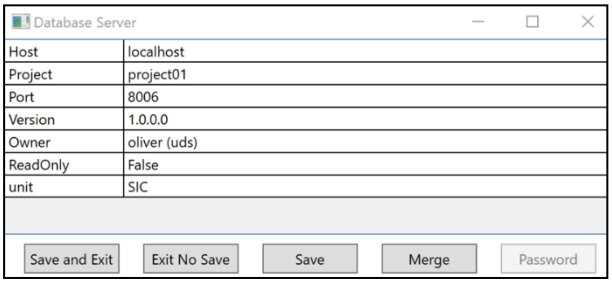

Fig. 3 Database server

3.1.2 Master Executive

The simulation model is controlled and run by a Master Executive as shown in Fig. 4 and 5. The Master Executive manages multiple executives (including the executable program of the simulation model). Each executive has a set of pages to execute. These pages are the modeling pages in the modeling and simulation workstation. Users can configure the execution rate of each page. A series of nodes in the page are modeling objects used to calculate simulation results as shown in Fig. 2.

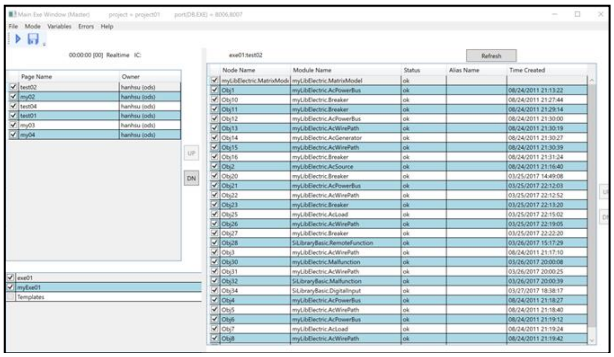

Fig. 4 Configuration of master executive

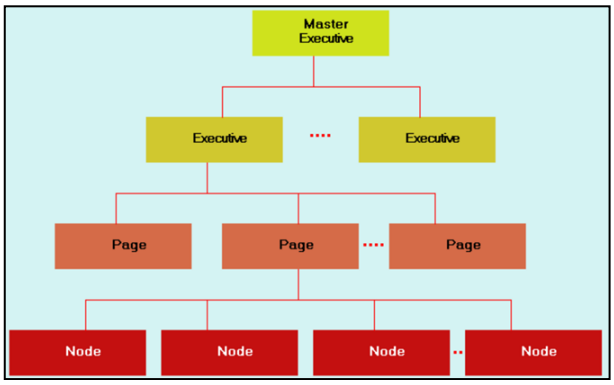

Fig. 5 Hierarchy of master executive

3.1.3 MBStation

MBStation is a Graphic User Interface(GUI)-based workstation as shown in Fig. 6 for users who operate or use simulation products. The graphics engine is a two-dimensional(2D) graphics editor. The editor provides many dynamic features for displaying station numbers based on animation needs. During startup, the workstation loads user-defined icons into the system from the list specified in the property file. The workstation has two modes: "Edit" and "Run". In the Edit mode, models are built page by page by dragging and dropping icons, and they are connected together through links. All icons and links are part of the modeling icon library. The interface between pages can be completed through virtual nodes. A virtual node is a node that represents any node on another page in the system. In the run mode, pages are animated by real-time data from executives. Users can observe the simulation results and change object data during the simulation process. Users start by creating a new page. Then drag and drop icons onto the canvas to create nodes for that page. Each page has attributes, including the name of the executive, the name of the page, the name of the subdirectory, and the cycle rate. Connections between nodes are made by selecting link icons and a series of mouse down/move/up operations. All of these editing operations are automatically registered with the Database Server and Master Executive. Users can edit modeling pages while in supervisor runtime, that is, online editing. MBStation

also includes the functions of an Instructor Station.

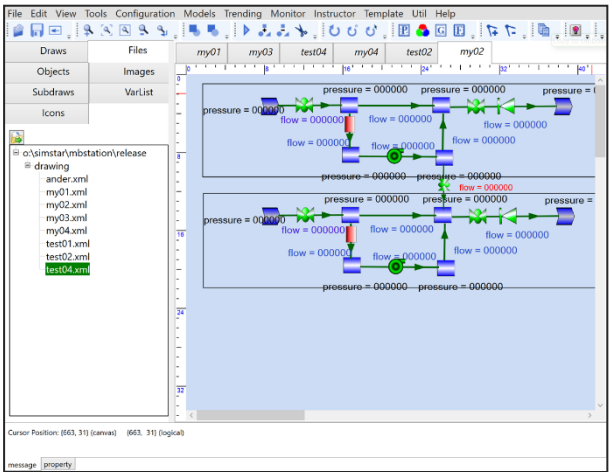

Fig. 6 Initial screen of MBStation

3.2 Code Integration Features

3.2.1 User Code Integration

Users use their code in a convenient environment provided by MBStation by separating and transplanting each element of the code they own into MBStation, and connecting the attributes and data to the icon that is basically installed in MBStation. This makes it easy to research or develop one's own in an integrated environment that links codes other than one's specialized field.

3.2.2 Icon Library

The icon is the smallest unit of a component in the system as shown in Fig. 7. The icon can be one of the following:

- Math module: used for mathematical calculations and saving related data.
- Graphical objects: Used for display on the canvas of the page. It includes many methods of editing and supporting dynamic display animations.
- Data object: Supports parameter editing for the above graphic objects.
- Display file: as a specified sub-graph object in the graph object.
- Image file: The image is loaded into the imagelist of the icon list view panel, and the icon will be selected from it.

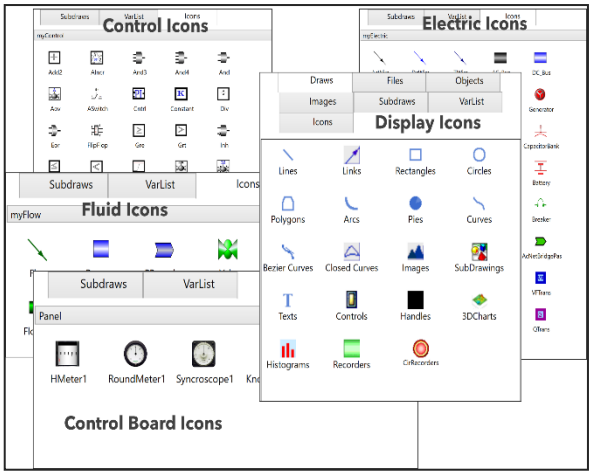

Fig. 7 The default installed icon library

4. Conclusion

In nuclear power plants, generating big data of nuclear power plants using modeling and simulation technologies is the core of technology development beyond human limits in the operation of nuclear power plants. To this end, MBStation was developed as platform software that conveniently generates various data in an integrated environment. This was developed to make it possible to accommodate codes in all fields of the nuclear power plant, and based on this, each user could further develop his or her field.

REFERENCES

[1] Domestic development and utilization of K-SMR digital twin platform technology, which integrates reality and virtuality, KNS Chosun Univ. Sungchul Lee, pp. 8, 2024.
[2] OpenSim Engineer's Manual, 2004. 6
[3] JADE™ Design Environment, 2010. 11
[4] 3KEYMASTER Analyst & Developer User Guide 2009. 5
[5] GES User Guide, 2009. 5