Simulation Program Algorithm for the Pyroprocessing Material Flow and MUF Uncertainty

Yos Panagaman Sitompul [a*], Hee-Sung Shin [a], Bo-Young Han [a], Ji-Sun Seo[b], and Ho-Dong Kim[a]
[a]Korea Atomic Energy Research Institute, 1045 Daedeokdaero, Yuseong-gu, Daejeon 305-353, Republic of Korea
[b]Hanyang University, 17 Haengdang-dong, Ansan, Kyunggi-do, Seoul 133-791, Republic of Korea
[*]*Corresponding author: yossitompul@kaeri.re.kr*

1. Introduction

An algorithm for the pyroprocessing material flow and MUF uncertainty simulation has been developed. The algorithm uses the threading method to modularize pyroprocessing system so that the process in the system can be modified and simulated independently of each other. The algorithm has been implemented in a simulation program called PYMUS (PYroprocess Material flow and MUF Uncertainty Simulation) which has been designed to be able to simulate the REPF (Reference Engineering scaled Pyroprocessing Facility) model and the future development of the model [1]. The threading method implemented in PYMUS gives users flexibility to design the model. The algorithm is described herein, and the result shows that the simulation program is flexible and realistic.

2. Methods and Results

In this section, the basic algorithm of PYMUS is explained briefly and some important aspects in the algorithm such as the process creator form, the threading method, and the MUF uncertainty estimation is explained specifically.

2.1Basic Algorithm

PYMUS' basic algorithm flowchart is shown in Fig. 1. To simulate the material flow in the pyroprocessing system, PYMUS uses the ORIGEN-S calculation to get the input nuclides and calculates the material balance for each process in the system [2]. A list of processes is given to PYMUS, including their information, such as the process formula. PYMUS follows the process list and imitates the batch process in the real situation.

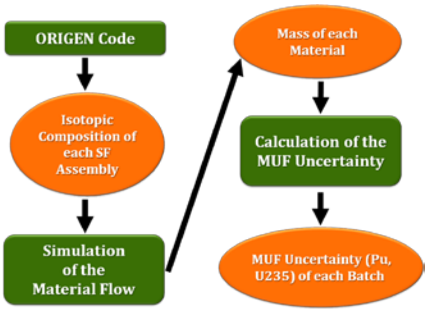

Fig. 1. PYMUS' basic algorithm flowchart.

For each batch, PYMUS gets the input material from the inventory, calculates the output material by using the formula, and stores the output material. These processes continue until all input campaigns are finished. All the getting and storing of the information are connected to the data base.

The MUF uncertainty is estimated by using the material amount in the KMP (Key Measurement Point) and the MUF method and stratum parameters that are preset up. The result can be shown by retrieving the data from the data base and showing them in form of graph or table.

2.2 Process Create Form

The program has been developed with flexibility for users to create their own pyroprocessing system. The users can edit the REPF model or start from scratch by adding, deleting, or moving some inventories, processes, materials, or lines in the system. This editing process is done visually by using the process creator form as shown in Fig. 2. The inventory, process, material, and line information can be added or deleted via this form. Users can setup the input by right clicking the object and selecting the option in the menu.

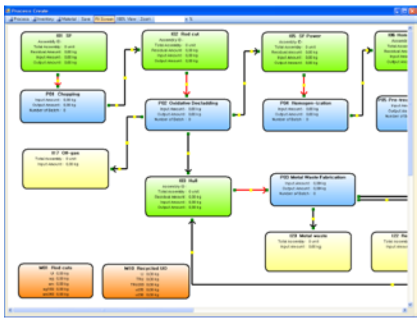

Fig. 2. Process creator form in PYMUS.

When the process creator form is activated, the user interface is loaded to the form. The form loads the data from the data base and displays it. The form has several buttons and menus which contain their own codes inside them. Most of them are connected to the data base. The users are actually changing the data base input by filling out the process creator form. When the save button is clicked, every works in the process creator form will be saved to the data base and loaded to the main window form.

2.3 Threading Method

The program has been developed to be modular. The users have a freedom to add and delete processes in the system. It also represents the near real time simulation. That means that the simulation is done based on the real time conversion of each process in the system. This thing can be done actually because each process in the system can work independently.

To simulate this independent process, the threading method was implemented in the program. In the threading method, each of process in the system is represented by the thread. Thread itself is the smallest unit of processing that can be scheduled by the operating system. It generally results from a fork of a computer program into two or more concurrently running tasks. Fork itself is a system call which is used by a process to create a copy of itself and both of them then run separately. By using thread, each of process in the system can run concurrently and independently.

In our algorithm, multiple threads exist and perform on a single processor. On a single processor, multithreading generally occurs by time-division multiplexing, the processor switches between different threads. This context switching happens frequently enough that the users perceive the threads or tasks as running at the same time (Fig. 3).

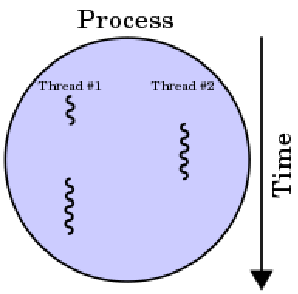

Fig. 3. Multithreading in single computer.

The flowchart of the threading method implemented in this paper is shown in Fig. 4. The main thread setup all input parameters required for the material flow simulation. The main thread then creates process threads from the process list and starts the threads.

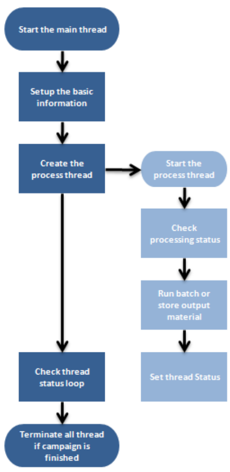

Fig. 4. Threading implementation flowchart in PYMUS.

Each process thread has a process as its argument. Each process thread has same instructions but works on different data. Each thread does not share memory because it has its own argument. To share data, each thread must give its data to the main thread and retrieve the other threads' data from the main thread.

When the users modify the process in the pyroprocessing system by adding, deleting, or moving, the users do not need to worry about the process order since each process works independently.

2.4 MUF Uncertainty Estimation

We provide input forms which are the MUF uncertainty method form and the MUF uncertainty stratum form for users to input the method and stratum parameters. By doing this, the users have a freedom to input the KMPs, methods, and their parameters. The formula for MUF uncertainty follows our latest MUF uncertainty estimation study performed by Han [3]. It includes the MUF uncertainty estimation from random and systematic error. Users can get the campaign MUF uncertainty result and total MUF uncertainty result from the simulation. The MUF uncertainty estimation result is shown in Fig. 5.

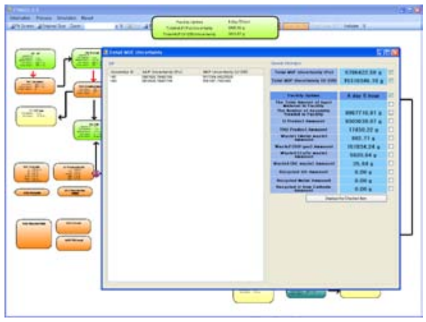

Fig. 5. MUF estimation uncertainty result.

3. Conclusions

The algorithm has been implemented in PYMUS. The threading method is the main key in the algorithm to make the program more flexible and realistic by modularizing the pyroprocessing system. The process creator form improves the designing approach by providing visualization. Users have a freedom to modify the REPF. The MUF uncertainty estimation algorithm is flexible so that the users can setup MUF stratum and method parameters freely.

REFERENCES

[1] H. S. Shin, B. Y. Han, S. K. Ahn, J. S. Seo, and H. D. Kim, Development of a Simulation Program for the Pyroprocessing Material Flow and MUF Uncertainty, INMM 52nd, California, USA, July 17-21, 2011.
[2] S. Bowman, "ORIGEN-ARP, A fast and easy-to-use source term generation tool," ICRS-9, Tsukuba, Japan, Oct 17-22, 1999.
[3] B. Y. Han, H. S. Shin and H. D. Kim, Evaluation of Material Unaccounted for Reference Pyroprocessing Facility Model, , INMM 52nd, California, USA, July 17-21, 2011.