A Study on Data Base for the Pyroprocessing Material Flow and MUF Uncertainty Simulation

Yos Panagaman Sitompul, Hee-Sung Shin, Bo-Young Han, and Ho-Dong Kim
Korea Atomic Energy Research Institute, 1045 Daedeokdaero, Yuseong-gu, Daejeon 305-353, Republic of Korea
yossitompul@kaeri.re.kr

1. Introduction

A simulation program called PYMUS (PYroprocess Material flow and MUF Uncertainty Simulation) has been developed to study the pyroprocessing material flow and MUF (Material Unaccounted For) uncertainty [1, 2]. In the program, there is a data base system that controls the data processing in the simulation. The data base system consists of input data base, data processing, and output data base. The data base system has been designed in such a way to be efficient. One example is using the OLE DB and MySQL. The data base system is explained in detail in this paper. The result shows that the data base system works well in the simulation.

2. Methods and Results

In this section, the data base system in PYMUS is explained. It includes the explanation of application program interface, input data base, data processing, and output data base.

2.1 Application Program Interface

PYMUS has been developed by using visual C++ program. There are two data base interfaces which are the OLEDB and MySQL. The reason for using two data base interfaces is to give an option for the users to choose the data base system suitable with the data base size. The OLEDB can work quickly but have a relatively small memory (2 Gb), while MySQL has infinite memory (limited only by hard disk size) but works slower than OLEDB. Both of them work by using the relational data base management system but only OLEDB that support object data base and spreadsheet such as Microsoft Access. The data base processing by OLEDB and MySQl are efficient because the data is stored and retrieved in a managable way. In practice, users can choose between the two interfaces.

2.2 Input Data Base

PYMUS has default input data base that contains the REPF (Reference Engineering scaled (10 ton/yr) Pyroprocessing Facility) data. The input data base is managed into several data base tables such as Element, BurnInfo, Display, FormulaInfo, FuelInfo, FuelNuclide, FuelType, GroupInfo, InventoryInfo, MufMethodInfo, MufStratumInfo, ProcessInfo, ProcessOutProductInfo, ProcessParameter, ProductGroupErrorInfo, ProductInfo, SimulatorConfig, and StartProductInfo tables. These tables are filled with the default data and are loaded automatically when the data base file is activated.

There are two ways to review the input data base and change it in PYMUS. First by using the input data form as shown in Fig. 1 and second by using the process creator form as shown in Fig. 2. Fig. 1 shows one of the input data forms in PYMUS. The actual number of input data forms is as many as the number of input data base tables. Users can select the item in the input data form and insert the input data. If the input data form is modified, then the input data base table corresponds to it is modified too.

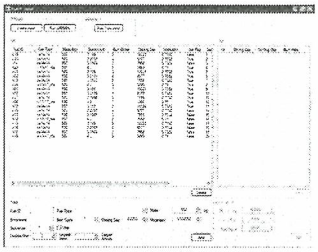

Fig. 1. Input data form in PYMUS.

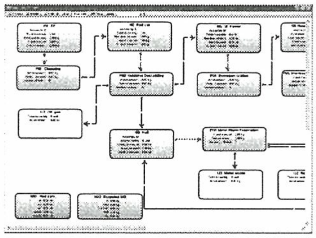

Fig. 2. Process creator form in PYMUS.

The process creator aids the users to setup the input data by providing visualization. The inventory, process, material, and line can be added or deleted via this form. The object in the form can be modified by right clicking it and selecting the option in the popup menu. Process creator form has several buttons and menus which are connected to the data base. The users can modify 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.

An important input data form to be described here is the FuelInfo input data form. The form contains information about the input spent fuel used in the simulation. In the form, there is an option to run ORIGEN-S calculation to produce the input nuclides for the simulation. By using the ORIGEN-S calculation, thousands of nuclides are produced [3]. The data base size is proportional to this number of the input nuclides. The nuclides are stored in the FuelNuclide data base table after the ORIGEN-S calculation is finished. During the simulation, the data base size increases. This increment is explained next in the data processing description.

2.3 Data Processing

To start the simulation, users need to make a data base selection first. Users need to choose between OLEDB and MySQL and then select the Ms. Access file if the users choose OLEDB (no need to select DB file if the users choose MySQL). Data base is automatically connected to PYMUS and default data base is loaded. This data base loading is managed by DBManager.cpp in PYMUS source code.

PYMUS uses the managed C++ which contains gcnew instruction. The gcnew instruction creates a managed reference in form of class. There are several classes such as simulator, process unit, inventory unit, product, etc. in PYMUS. These classes are managed by using a list, mapping list, or dictionary. These managing instructions are well documented in MSDN website.

The first data base loading is actually retrieving all the input data from the input data base and copy them to the managed reference by using a list, mapping list, or dictionary. The code example of the data base loading is shown in Fig. 3.

```
MappingList<InventoryInfo^>^ MDBDBManager::getInventoryInfoList()
{
        MappingList<InventoryInfo^>^ retList = gcnew MappingList<InventoryInfo^>();
        OleDbDataReader^ dr = Query(L"select * from new_InventoryInfo ORDER BY inventoryID ASC");

        while(dr->Read())
        {
                InventoryInfo^ info = gcnew InventoryInfo();
                info->inventoryID = dr[L"inventoryID"]->ToString();
                info->inventoryName = dr[L"inventoryName"]->ToString();
                info->isResult = Convert::ToBoolean(dr[L"resultFlag"]->ToString());

                retList->addItem(info->inventoryID,info);
        }
        dr->Close();
        delete dr;

        return retList;
}
```

Fig. 3. Code example of the data base loading.

After the data base is load, PYMUS uses the reference for the calculation. In the batch processing, there are output material calculation and output material storage to inventory. In PYMUS source code, in the output material storage to inventory subroutine, there is a subroutine calling called AddLogOut to add the information of the output material to the output data base. The output material storage to inventory subroutine happens many times during the simulation so that the data base size increases overtime. The data base size depends highly on the number of output materials which are actually depends on the number of input nuclides. To reduce the data base size, we give an option in the input data form to filter the input nuclides by selecting the nuclides based on their masses and activities.

2.4 Output Data Base

The output data base is needed when the users want to retrieve the simulation history, for example, when the users want to check the total output material produced by one process. The nuclides from output data base are retrieved for this purpose. The output material information can be check by double clicking the item box in the simulation. One example is shown in Fig. 4. PYMUS provides also graphs and tables to report the simulation result. All result graphs and tables use the output data base. One graph example is shown in Fig. 5.

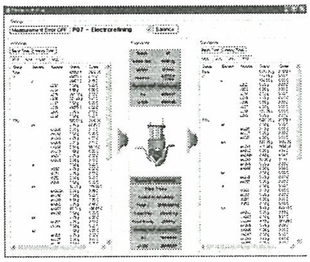

Fig. 4. Process batch information in PYMUS.

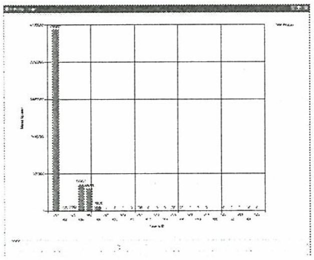

Fig. 5. Product result graph in PYMUS.

3. Conclusions

The data base for the pyroprocessing material flow and MUF uncertainty simulation has been implemented well. There is no error in the data base processing and it is relatively fast by using OLEDB and MySQL. The important issue is the data base size. In OLEDB the data base size is limited to 2 Gb. To reduce the data base size, we give an option for users to filter the input nuclides based on their masses and activities.

4. 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] 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.

[3] S. Bowman, "ORIGEN-ARP, A fast and easy-to-use source term generation tool," ICRS-9, Tsukuba, Japan, Oct 17-22, 1999.