Method for Database-oriented Integrated Environment of Chemical Engineering Software

Jianhua Yu

Academy of Innovation Education, Chongqing Technology and Business Institute, Chongqing 401520,China
jianhuayu2314@126.com

This paper aims to verify the validity and reliability of database-oriented method in the integrated environment of chemical engineering software. To this end, a comprehensive analysis for the existing integrated environment method of chemical engineering software and its actual use was conducted, and then by combining the characteristics of Windows with the conventional methods in the management of application programs, the study was conducted for the integration strategy of software application in the system environment. The results show that in the integrated environment of chemical engineering software, the application of database can be used to establish special software, enabling the software application function to be more perfect with higher feasibility and reliability. It's concluded that in the integration of chemical engineering software, through the data establishment, it can have the functions of other software, and one dedicated database management system for chemical engineering software is also established. Therefore, in the integrated environment of chemical engineering software, the application of databases plays a key role.

1. Introduction

With the promotion of social economy, the computer technology has been applied continuously in more fields, and it has gradually become the key technology for achieving sustainable development in all walks of life. Under this background, new software is constantly emerging. Among these software, some are developed only to solve one certain type of problem, usually, several software need to be used repeatedly to effectively solve a chemical problem. In view of the current computer development situation, it is difficult to realize the performance of other software on the basis of one software, and it is also not economical in this way. Thus, software integration has emerged as a new type of software development in the current social and economic development process, which has attracted the attention of scholars in many fields. At present, the software integration technology mainly includes software bus technology, OLE technology, and Activex container technology; for the latter two methods, in the actual application process, there are certain limitations in the types of integrated software. Compared with the remaining integration methods, relevant research results aren't also mature enough.

Therefore, this paper proposes to implement integration strategies for independent software in its system environment by using the advantages of Windows. Besides, in this integrated environment, four modules were established, namely, database management subsystem, code integrated module, interface integrated module, and data integrated module. Finally, the feasibility and process of these four modules applied in the integrated environment of chemical engineering software were explored in this paper.

2. Literature review

The wide application of computer technology has greatly promoted the management and technological progress of all walks of life. The resulting software industry has shown a booming trend. Hundreds of general and special software have been introduced, and their performance and quality are quite high. However, these softwares are only developed to solve certain problems. For an actual chemical problem, it is often necessary to use several software repeatedly to solve it. The ability to develop other software on a software basis is

neither economical nor easy to accomplish. As a new software development method, software integration has attracted more and more attention of scholars.

The factors that influence the quality of the optimized software were analyzed and studied comprehensively (Mitkowski and Bal, 2015). The ultimate goal of optimizing software is to solve large-scale practical engineering optimization problems. Optimization software must meet requirements such as robustness, extensibility, and user interface optimization. The richness of optimization algorithms contained in the optimization software is also an important indicator for evaluating optimization software (Arnot et al., 2015). Curtis et al. discussed the hardware and software computing environment for optimizing software (Matsuo et al., 2017).

Some scholars have developed the advanced interface software of Aspen Plus software and Matlab software, namely the MAP toolbox. The toolbox is based on COM technology and follows the structure of the Aspend Plus data browser. An interface program was written between the two software. The powerful computing power of Matlab software is combined with the superior process simulation capabilities of Aspen Plus software. The control of running Aspen Plus software under Matlab environment is realized. The parameters of the Aspen Plus model can be modified in the Matlab software. After the simulation of the model is converged, the simulation result can be read and written into the specified position. The MAP Toolbox also extends the simulation capabilities of the Aspen Plus software. For example, by adding advanced algorithms, it solves the problem of optimization that is difficult for Aspen Plus (Grondin et al., 2016).

The Aspen Plus user interface is an ActiveX server application. Based on this, it is used as an interface with external applications. The interface software was developed in the VB (Visual Basic) environment, and the objects provided by the Aspen Plus can be accessed in the VB programming environment to implement control of the Aspen Plus. Taking a toluene absorption tower as an example, three variables were studied: the flow of toluene in the absorbent, the position of the feed plate, and the position of the intercooler. With toluene as the target for the absorption of key components, 100 sets of variables generated randomly in VB were passed to Aspen Plus for simulation calculations and the simulation results were read back to VB for storage. This avoids manual input of variables in the Aspen Plus interface to obtain simulation results. It provides assistance for large-scale simulation calculations and avoids the cumbersome manual operation. However, this method is randomly generated when the variables are assigned and passed to Aspen Plus in turn. It is difficult to obtain a globally optimal operating variable (Andersen et al., 2016).

Using Excel's own Visual basic for applications (VBA) programming function, Excel software and PRO/II software interface programs were written. In Excel software, the control of PRO/II software operation and data reading and writing are implemented. The design of the rectification column of benzene-toluene binary mixture was taken as the research object, and the relationship between the number of theoretical plates and the reflux ratio was studied. At the same time, the relationship between operating costs, equipment investment costs and return flow ratios was studied. In VBA, the PRO/II software was called to perform simulations and hydraulic calculations at different plate counts. The reflux ratios, investment costs, and operating costs of the fractional distillation column under different theoretical plate numbers were obtained. Then, the simulation results are written into Excel software. By using Excel's good data processing capabilities, investment costs and operating costs are weighed. In the end, the number of plates and the reflux ratio are optimized. This method improves the design efficiency of the rectification tower. However, the lack of optimization filtering for variable assignment makes it difficult to obtain optimal operating variables. These studies basically use Excel's good data processing capabilities to help improve the computational efficiency of PRO/II software or Aspen Plus software. Few researches are about the multivariate optimization problem of complex chemical system processes. This is exactly what is needed in the actual industrial process (Zhou et al., 2016).

A multivariate optimization study was conducted. Using the Aspen Plus software's sensitivity analysis, a large number of simulation calculations of the research object were obtained, and the mapping sample data of each variable and the target parameter was obtained. Through polynomial analysis, the data is simulated and the functional relationship between the target parameters and each variable is fitted. With data mining, the engineering relationships between parameters are found. Based on empirical equations and design data, the cost function of the process is determined. The minimum cost function is the goal of optimization. Genetic algorithm and pattern search are used to calculate the optimal solution. All work is done manually offline. AspenPlus software and Matlab software are not automatically connected. Therefore, the limitations are greater (Caspi et al., 2017).

Based on the CAPE-OPEN technology, the integration of various process simulation software was studied. "Co-Simulation" was proposed. For example, a joint simulation of crude oil extraction and refining links is used to guide the production operations of the oil recovery and refining processes. However, the method emphasizes the invocation of different methods for different physical properties of different software, and does not study the parameter optimization of chemical process simulation (Raies and Bajic, 2016).

In summary, by studying the connection function between the PRO/II COM server and other software, an interface program of PRO/II software and Excel software was written in VBA. At the same time, in the program, the genetic algorithm is written. The simulation capabilities of PRO/II software, the data processing capabilities of Excel software, and the optimization capabilities of genetic algorithms are integrated. This can handle multivariate optimization problems for complex chemical processes. Then, a multi-objective genetic algorithm program was programmed in the interface program to realize the multi-objective optimization function. The Pareto solution set under multi-objective can be solved. A case study of fractionation system in a refinery gas catalytic cracking unit shows that the method is feasible.

3. Methods

The module level of the object-oriented software integrated environment was designed. At this module level, an object-oriented software integrated environment (OOSIE) was preliminarily designed. The object refers to an entity that contains certain characteristics and has a specific behaviour. Conceptually, it is a user-definable data type. It itself contains some data types to describe its own characteristics, and also some functions to describe its behaviour so that users can operate on it. Even an object can include other objects. It has three basic characteristics: encapsulation, inheritance, and polymorphism. In this integrated environment the object-oriented technology is integrated into different stages of software analysis, software design, software generation, and software maintenance etc, enabling the users to ultimately generate software products that meet software engineering standards on this platform. At present, the development of domestic system software is mostly in the manual stage. At the stages such as system sub-bridge, design, programming, document generation, and software testing, there is large demands for manual intervention, which greatly reduces the development efficiency, and also makes it difficult to achieve the results in line with the software engineering standards. Besides, the use of object-oriented technology in this integrated environment can significantly increase the degree of automation for software development and greatly reduce the amount of software maintenance. Figure 1 depicts the design strategy of integrated environment in this paper. Four major functional modules were designed for this integrated environment: interface integrated module, code integrated module, data integrated module, and database management subsystem.
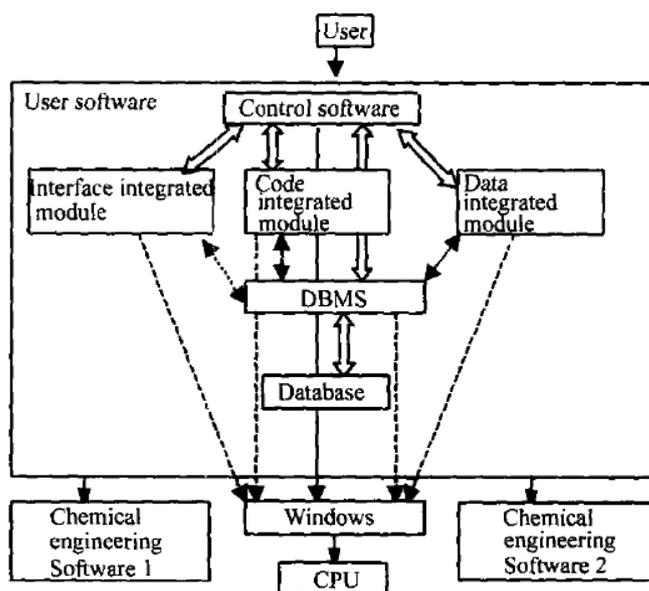
Figure 1: Research strategy of chemical engineering software integration environment

4. Results and analysis

4.1 Features of integrated environment

In the integrated development environment of visual C++6.0, this paper studies the integration of chemical engineering software under Windows system environment, and then designs the overall structure of the chemical engineering software integration environment. This integrated environment supports secondary

development. The integrated environment has also an independent database and data-driven database management subsystem. For chemical engineering software, data processing is its main function, because most software has a separate database for storage of various physical information, optimized parameters, and intermediate processing results etc. and the establishment and management of various software databases are various.☐

3.2 Interface integrated module

3.2.1 System support for interface integration

Windows programs are based on messages and driven by events, while messages are managed by Windows. There is no exception to all chemical engineering software under the Windows system. Message management mechanism of Windows is shown in Figure 2.

3.2.2 Interface integration strategy

User interface integration refers to the interface control of each integrated software in the integrated user software. There are two methods of interface integration. One is to connect or embed the integrated software interface into the user software, and then the user switches to use the required software interface in the integrated software. If the integration effect of these two methods is expressed by an equation, that is, the former method is "1+1=2", and the latter is "1+I>2". For the first method of interface integration, the current more mature ActiveX container technology, OLE technology and other software integration technology can be adopted by using the connection and embedding method to integrate the ActiveX control, ActiveX document and the program interface of documents to the soft sample of user. This integrated interface facilitates the user to operate in a familiar environment. For the second method of interface integration, Windows message and object Management mechanisms can be used for integration. Most object handles in an application are reserved and controlled by Windows.
Data integrated module
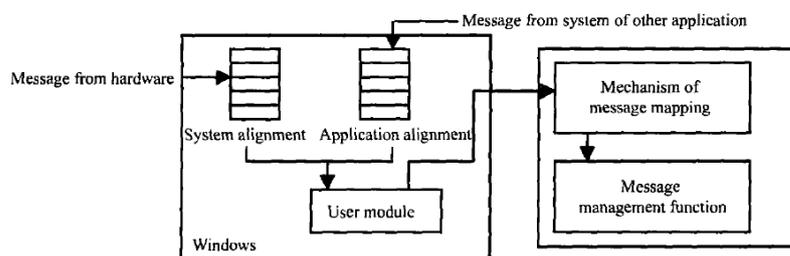
Figure 2: Message management mechanism for Windows

3.2.3 System support for data integration

Table 1: Operation mode of common controls

The control of	Control class name	Mode of operation
Static text control	Static	No mouse keyboard operation, can read
Edit control	Edit	Receive keyboard messages
The button control	Button	Accept mouse messages,
Check Box	Button	Accept mouse messages,
Radio Box	Button	Accept mouse messages,
Composite edit box control	Comb0B0xxx	No mouse keyboard operation, can read
Listbox control	ListBox	Accept mouse messages,
Scroll up and down control	ScrollBar	Accept mouse messages,
Left and right scroll-control software	ScrollBar	Accept mouse messages,
List contro	SysTabCofltrol132	Accept mouse messages,
Rich Edit	RichEdit20W	No mouse keyboard operation, can read

The 32-bit operating systems developed by Microsoft such as win95, win97, wings, and winNT series, are all multi-user and multi-task operating systems. Users can run multiple applications at the same time, while the

applications have their own virtual storage space and are independent of each other. However, if it's required to develop multiple applications and integrate them together to complete specific functions, various applications should call such functions to manage data exchanges between applications. The size of clipboard is variable, which can take up the entire effective memory space of the computer. There are a lot of data formats that the clipboard can use and accept, and more than 20 constant formats are defined in the window and APx for the user's selection. The user program can realize the transfer of special types of data by defining the required clipboard format such as the class object etc., e.g., through the text, graphics, OLE controls, and other types of messages sent according to the different types of control types, the problem can be resolved. Table 1 lists the relevant operation modes of various common controls and its class name.

3.2.4 Data integration strategy

Data integration means that data is shared and automatically invoked in each integrated software. This paper uses the system's support for the clipboard mechanism to implement data sharing and transfer between chemical engineering software (software 1 and 2) in the data integrated module. Figure 3 shows the implementation strategy:
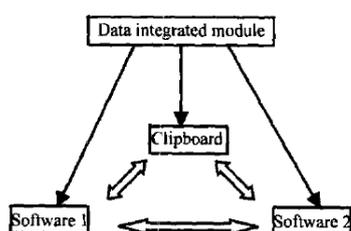
Figure 3: Data integration strategy

3.3 Database establishment

For chemical engineering software, data processing is its main function. For the four modules that make up the integrated environment, its functionality is also centred around the data. Therefore, in the chemical engineering software integrated environment, one data-driven database and its management system should be established. At present, the relational databases such as DBASB and FOXBASE are commonly used on microcomputers, but it is not appropriate to use them for data-driven strategy. This is because they occupy relatively large amounts of memory, and it is not convenient to mix them with high-level languages such as FORTRAN and C, esp., the relational data model is inconvenient to express the data involved in chemical process simulation calculation. Based on this, after analysing the data involved in chemical process calculations, the hierarchical and relational database model was proposed. Table 3 lists the main data objects.

Table 3: Data main objects

object	describe
Hapl,IP	Customer object
IHNode	Input and output data on the tree structure diagram
IHNodeCol	Each object can have its own other nodes, these Nodes are organized in a collection object
IHAI,Engine	Simulation engine provides an interface,

Chemical steady-state simulation may have the following tasks: process simulation, condition analysis, process design, process optimization, equipment calculation, chemical data processing, and economic evaluation. The data involved in these tasks can be divided into two parts: the data with one-to-one correspondence, and those with one-to-many or many-to-many relationship; the latter is a portrayal of complex links between logistics and equipment, between equipment. Relative to specific logistics and equipment unit parameters, they are referred to as system-level data.

The hierarchical model involves the index name of each module and logistics, and the corresponding logistics information and equipment parameters need to be stored in the relational database. In addition, experimental data and experimental points, iterative results and iteration times should also be described in the relational model.

This paper establishes one data-driven database management subsystem for data-driven databases. In addition to the functions such as addition and modification of data owned by the ordinary database management system, the sub-system also has the function of calculating the physical properties of the data and optimizing the calculation according to the user's needs.

4. Conclusions

This paper studies the integration of chemical engineering software under Windows system environment, and also designs a chemical engineering software integration environment composed of interface integrated module, code integrated module, data integrated module and database management subsystem. Then, it describes the functions of the four modules that make up the integrated environment, and proposes interface integration strategy, code integration strategy, data integration strategy, and the establishment method of data-driven database and database management subsystems.

With the ever-increasing data, research on improving its functionality and applicability in different field has become a research hotspot for many researchers at home and abroad, and a large amount of research work is being carried out from various perspective. Due to the limitation of the research basis and time, the author has only conducted preliminary research and exploration on the database module, and there is still a large amount of research work to be conducted.

Reference

Andersen J.H., Murray C., Larsen M.M., Green N., Høgåsen T., Dahlgren E., 2016, Development and testing of a prototype tool for integrated assessment of chemical status in marine environments, Environmental Monitoring and Assessment, 188(2), 115, DOI: 10.1007/s10661-016-5121-x

Arnot J.A., Quinn C.L., 2015, Development and evaluation of a database of dietary bioaccumulation test data for organic chemicals in fish, Environmental Science & Technology, 49(8), 4783-4796, DOI: 10.1021/es506251q

Caspi R., Billington R., Fulcher C.A., Keseler I.M., Kothari A., Krummenacker M., 2017, The metacyc database of metabolic pathways and enzymes, Nucleic Acids Research, 42, 459-71, DOI: 10.1093/nar/gkm900

Grondin C.J., Davis A.P., Wiegers T.C., King B.L., Wiegers J.A., Reif D.M., 2016, Advancing exposure science through chemical data curation and integration in the comparative toxicogenomics database, Environmental Health Perspectives, 124(10), 1592, DOI: 10.1289/ehp174

Matsuo T., Tsugawa H., Miyagawa H., Fukusaki E., 2017, Integrated strategy for unknown EI–MS identification using quality control calibration curve, multivariate analysis, EI–MS spectral database, and retention index prediction, Analytical Chemistry, 89(12), 6766-6773, DOI: 10.1021/acs.analchem.7b01010

Mitkowski P.T., Bal S.K., 2015, Integration of fire and explosion index in 3D process plant design software, Chemical Engineering & Technology, 38(7), 1212-1222, DOI: 10.1002/ceat.201400644

Raies A.B., Bajic V.B., 2016, In silico toxicology: computational methods for the prediction of chemical toxicity, Wiley Interdisciplinary Reviews: Computational Molecular Science, 6(2), 147-172, DOI: 10.1002/wcms.1240

Zhou D.D., Xu, Y.T., Lin R.B., Mo Z.W., Zhang W.X., Zhang J.P., 2016, High-symmetry hydrogen-bonded organic frameworks: air separation and crystal-to-crystal structural transformation, Chemical Communications, 52(28), 4991-4994, DOI: 10.1039/c6cc00366d