The Research of Plug-in Based CAPP System with plug-and-play Characteristic

ZHANG Sheng-wen， LIU Jin-ju， FANG Xi-feng
(Modern Manufacturing Technology Institute, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003, China, E-mail: ljju123@163.com)

Abstract

It is difficult to solve the flexibility and adaptability of a computer aided process planning (CAPP) system. This paper analyzes the disadvantages of researching methods in traditional CAPP systems and presents the advantages and characteristics of Platform/Plug-in structure, utilizes the Platform/Plug-in software structure to develop the framework of CAPP system. Using Plug-in technology to realize the main functions of CAPP system made the system excellent in plug-and-play characteristic. The presented paper discusses the basic thought of Plug-in technology, and establishes a framework of a Plug-in based CAPP system, provides the steps of developing process, accomplishes the customization development of the key functions and the assemble on the platform. By using the Plug-in technology with plug-and-play trait, the function of the system is improved without changing the main program. It significantly improves the flexibility and reusability of the system and it has great effect on the adaptability in CAPP domain.

Key words: Plug-and-Play; CAPP; platform/plug-in software structure; Plug-in technology

1. Introduction

The development of a CAPP system always followed the stereotype of developing special systems aiming at a single corporation, a single product and the single producing type, each time new CAPP systems from scratch have to developed, which cause difficult development, long period, high cost, and the system has poor scalability and is hardly be reconfigured and reused which lagged the popularization and application of system [1].

To this problem, the traditional ways to improve the adaptability and efficiency were from reasoning strategy and making CAPP developing tools, etc [2-4]. The paper presents the idea to utilize the existing software technologies and excellent software architecture to improve the CAPP system's flexibility and adaptability.

2. The Elementary Principle of Plug-in Technology

2.1. The essence and main characteristics of plug-in

The essence of plug-in is [5]: the function of the system can be extended and improved without changing the main program (platform), any company and any person can build its own plug-ins to solve the inconvenient of operating or add new functions for the system when needed. A framework of a plug-in includes two parts: main program (main-app) and plug-in. The main program is a program that contains plug-ins. A plug-in must implement some standard interfaces used by the main program when it communicates with the plug-ins, which made the software development have the real plug-and-play characteristic.

From the principle aspect, a main program, which was used as a container for plug-ins, ought to have the following main characteristics [6]: Firstly, the main-app should identify and load the plug-ins automatically. Secondly, it can build user interfaces, related to plug-in module dynamically, according to the requirements of the system, therefore the main-app can trigger and use the function inside. Adding elements to the system's menu bar and tool bar were the common ways. Thirdly, the main program of plug-in must establish rules (the interface of information exchange) for plug-ins to observe when they need to communicate with each other, ensuring plug-ins can exchange information and

978-0-7695-3570-8/09 $25.00 © 2009 IEEE
DOI 10.1109/WCSE.2009.464

triggered by the main program. Besides, different plug-ins of diverse functions had different requirements. Strictly speaking, plug-in was just a conception, not a principle, the above three were just established by usage, but not every application support plug-ins must implement completely, the developer can create its own trait according to the special requirements.

2.2. The framework and advantage of plug-in

The main program, based on plug-in structure, included four parts (figure.1) [7]: the framework of application, plug-in interface, plug-ins and public function library. ① the framework of application takes charge of the whole course of the program, manages the entire flow of the program, but it doesn't know what to do at every process, instead it triggers the plug-in to perfect the functions at the pudding time. ② plug-in interface is a protocol, which may be described in IDL, located in the head-files, or described in a paragraph of words. If only the plug-ins followed the protocol, they can be used in the main-app. Of course, there may be lots of plug-in interfaces for a complex system and each one had their own functions.③plug-in is the entity which realizes the specific function and implements the required plug-in interfaces. Although what function to realize and how to realize them are completely free, actually, there are some limits, because the plug-in interface itself is also a rule, such as a plug-in used as a compiler can't be used as a chatting tool. ④public function library, which is a group of functions or classes and can be used by the application framework and plug-ins, is usually a dynamic link libraries absolutely(DLL). The application framework itself is a public program which can be reused in code degree. But not all reusable code can be put into the framework, especially the public code is used by plug-ins, that may make plug-in have great dependence upon the framework. Putting those public codes into a standard library is a good way for reuse. Generally, the process of adding plug-ins was registered in configuration file which is saved in register file or configure file, It can be enrolled into the file dynamically or copied into the given location by the clients.

The advantages of using plug-in framework in software developing include: ① it reduced the complexity of developing the framework and made the framework more easily to realize by separating the extended functions;②the extended functions and the framework were coupling in a loose way, they can be exchanged separately and released when the interface was unchanged;③the other program can extend the function of the system when the plug-ins' interfaces were opened.

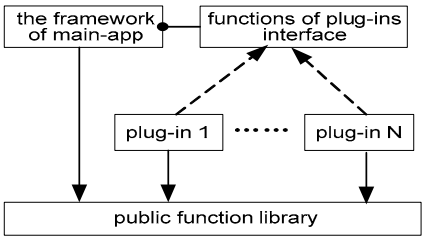

Figure 1. The framework of plug in program

3. The Framework of Plug-in based CAPP System

3.1. The Architecture of CAPP system based on platform/plug-in structure

The whole system was developed in plug-in ways, accordingly we must choose the corresponding software architecture to construct system. The paper firstly adopts platform/plug-in software structure in the building of CAPP system (figure.2). After analyzing of some CAPP software in detail, the key functions of the CAPP system has been divided, then we implemented it on the platform, and the less important functions were developed in the plug-in ways. At last, all of the plug-ins were assembled on the platform, therefore, an extendable CAPP system with plug-and-play characteristic comes into being.

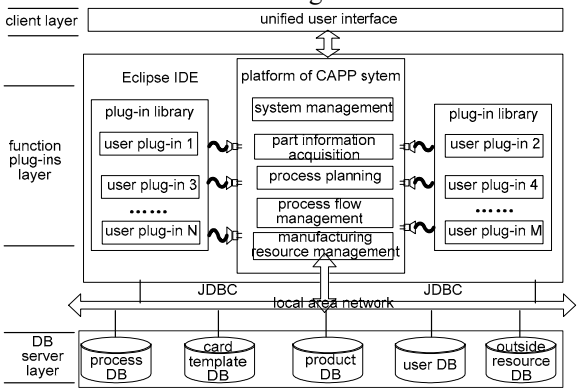

Figure 2. The framework of plug-in based CAPP software

Research shows that, a CAPP system commonly includes the following modules [1,8-9]: part information acquisition, process planning, process flow management, process file management, manufacturing resource management and system management. Because the process planning is the most important module for a CAPP system, it was implemented on the platform; the other general functions were developed as plug-ins which composed

a common plug-in library. For the other functions, which are not required by every corporation or every department, such as the computing of time using, the statistic of materials, were also developed in plug-in ways, which composed a special plug-in library. The company can use those plug-ins conveniently when needed, it can lessen the developing process and save lots of manpower. Therefore, using the platform/plug-in software structure can enhance the CAPP software's lifecycle.

（1）The client layer

By providing friendly unified user interface, clients can make the process planning under the guidance of the visible interface.

（2）The function plug-ins layer

The application program, which is used for storage of various plug-ins and platform of CAPP system, is the organization of system resources and implement layer of the function. From one aspect, it accomplished the customization of the system's required function and data; from the other aspect, it assembled the platform and plug-ins together, then clients can use the system for process planning.

（3）The DB server layer

It is used to save lots of data: the knowledge of the products, process data, equipment data, some universal manufacturing resource data and the special resource data for the given corporation.

3.2. The development process of Plug-in based CAPP system

The system selected Java as the programming language and Eclipse IDE as the developing tool, this plug-in based CAPP system fully used the Eclipse platform's structure and thoughts ----all were plug-ins. Figures 3 and 4 were the states of Eclipse workbench during the development period.

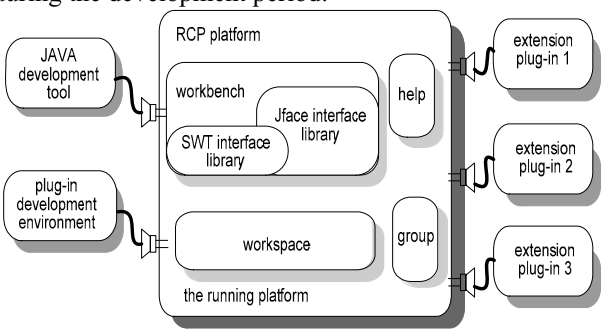

Figure 3. The state of Eclipse workbench before developed

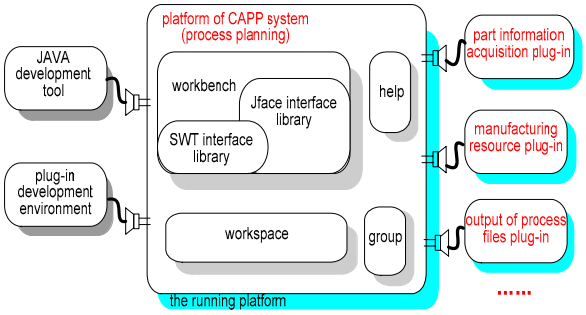

Figure 4. The state of Eclipse workbench after developed

4. The main functions of plug-in based CAPP system and their implement

The main processes in the developing of plug-in based CAPP were: the implement of process planning platform, the developing of various plug-ins and then assembling the plug-ins on the main platform. Except the process planning platform, the other functions used plug-in technology as the developing ways during the developing process. The author selected some general functions as practical example to realize (figure.5): part information acquisition, process flow management, manufacturing resource management and the output of process documents, as the limited size of the paper, we just took one of the plug-ins------output of process documents as an example to introduce.

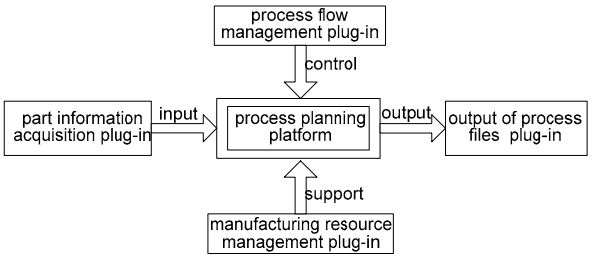

Figure 5. The relations of various plug-ins in plug-in based CAPP system

4.1. Process planning platform based on RCP technology

For a software system, which built in platform/plug-in structure, the platform must have the ability to recognize and load various plug-ins automatically. During the development of the process planning platform, the system fully utilized the Rich Client Platform technology of Eclipse workbench, which can build the platform conveniently, the process planning platform was the microkernel of the whole CAPP system, which was the mother of other

functions, built in plug-in methods, and it took great important in the running of the system.

（1）The load of the plug-ins for the platform

The trait of loading the plug-ins was an easy function for a plug-in based system. According to the loading time, it had the following ways: bundle at loading, bundle at running and the minimum load methods of Eclipse workbench, and so on. The basal principle was: the main-app set a path, then users copy their plug-ins and set them into the file below the path, the main-app will visit the directory at proper time to call the plug-ins and load them automatically.

In this system, the main-app was developed on the Eclipse RCP workbench and it had the advantage of using the minimum load methods of Eclipse workbench[10-11]. After it was developed, the platform had been released as a CAPP product(figure.6), clients first recorded the ID num of plug-in in the configuration files named cofig.ini, then copied the plug-ins' .jar file and placed them under the file named plugins, at last the user can double-click the .exe file to start the system and then can see all of the recorded plug-ins being loaded on the main platform.

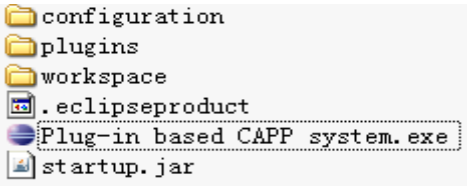

Figure 6. The product of plug-in based CAPP system

（2）The internal communication between plug-ins

The internal communication between plug-ins was the most important part for a plug-in based system, the main-app and plug-ins must extend each other's extension points (figure.7) to realize their functions[12]. During the designing period, We must compute the coupling degree of plug-ins, if the degree was too large, then plug-ins must communicate frequently, that enhanced the difficulty of developing; Otherwise, the system is too separate and loose to contact with others, therefore weak the relation and decrease the reusable degree.

The process planning platform, which developed on the Eclipse RCP workbench, provided lots of opening extension points. In order to communicate with the others, the plug-ins only need to extend those points and use the functions, this characteristic provided a steady theory foundation for the latter increment developing process.

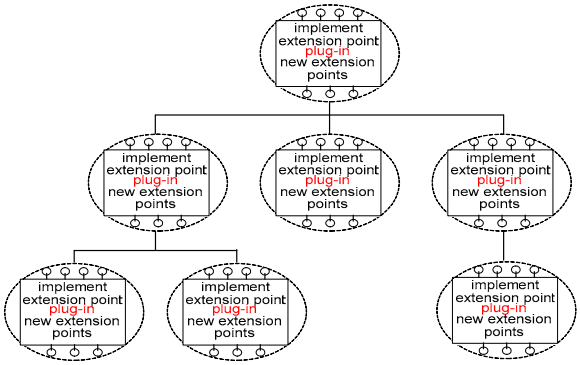

Figure 7. The extension points of plug-in

（3）Adding elements on the main frame of platform

There were two ways of adding elements to the main frame: one was to allot the content to the main-app to realize it when it was initialized; the other was exposed the interfaces of the platform to the plug-ins and the main-app just need to call the function of the interfaces.

In this system, all of the functions were developed as plug-ins, we choose the second adding ways, each plug-in had its menu bar and tool bar points, we just need to extend those points on the platform and trigger them by the elements, which were added on the platform by plug-ins themselves.

4.2. Example of output of documents based on iReport

The plug-in project was created on a plug-in template of Eclipse workbench, the programmer just need to instantiate those entities of classes to realize the particular functions during the developing period.

In the developing period, the plug-in of outputting the process documents mainly analyzed the types of the process documents based on product CAPP system, proposed to use the iReport as the report tool in the report designer. In the outputting plug-in, we implemented the display, the preview, the print and the setting of page of the documents based on the various database, and this plug-in was one of important tools for the CAPP system.

5. Conclusions

The basic thoughts of Plug-in technology was discussed and the steps of developing the CAPP system were given, which enhanced flexibility and adaptability of CAPP system, reduced the cost of developing and maintenance. Many advanced design concepts such as extreme programming ("XP" principle), increment software developing, object-

oriented programming were introduced. By using the platform/plug-in architecture in software designing, it made the CAPP system have the characteristics of assembling, deleting and adding plug-ins of various functions automatically, this kind of component design made the system have the plug-and-play characteristic. This new method made the functions of the system much easier to extend and broke a new path for the application of the CAPP system.

References

[1] Ke Wen and Wang Ning-sheng, "Research of Component Reuse in CAPP Domain"(graduate paper), *Nanjing University of Aeronautics and Astronautics*. 2003.9

[2]Chun Zhang and Shaw C.Feng, "An Architecture of Component-based CAPP System for Agile Manufacturing", *The National Institute of Standards and Technology(NIST)*. 1996.

[3] T.H.M. Nguyen, J.R.Duflou and J.-P. Kruth, "A Framework for Automatic Tool Selection in Integrated CAPP for Sheet Metal Bending", *Advanced Materials Research*, Vol.6-8(May 2005)pp.287-294.

[4] Kamrani A K and et al, "Critical Issues in Design and Evaluation of Computer Aided Process Planning System", *Computers and Industrial Engineering*,1995,29(1-4) pp:619-623.

[5] LI Yan-chun., "Theory and Implementation of Plug-in Technology", *Computer Systems & Applications*, 2003,7 pp:24-26. (in Chinese)

[6] Tony Million, Al-Riaz Adatia and Andrew McCann, "Nicholas Vinen.Secure flexible plugin software architecture", *United States Patent Million et al,* May 25,2004.9 pp:572-593.

[7] Mark Brown, SHI Fu-yuan and SHI Ya-wei, *WWW Plug-Ins Companion*,Que Corporation. 1997.7

[8] WANG Yun-li, DUAN Guang-hong, LIU Dan and MAO Wei, "Research on Key Technologies of User Frame System in CAPP Developing Tool", *CIMS*. 2000.2. 6(1) pp:25-30. (in Chinese)

[9] M. HOU and T.N. FADDIS, "Automatic tool path generation of a feature-based CAD/CAPP/CAM integrated system", *International Journal of Computer Integrated Manufacturing*, 2006.6.19(4) pp:350-358.

[10] Bruce Eckel, *Thinking in Java*, China Machine PRESS. 2004.2.

[11] CHEN Gang, *Mastering Eclipse (The 2ed edition)*, Press of Tsinghua University. 2007.7.

[12] LIU Yi, CHEN Hui-rong, RONG Gang. Program Design of Plug_in Structre. *Computer Application.* 2002.4.22(4)pp:9-12.