Design and Implementation Method for Supporting Complex CAE Software System Module Development

Xiaohui Wang 
Xiaohua Nie 
Liang Chang 
Shihui Duan

1 Aircraft Strength Research Institute of China, Xi’an Shanxi 710065, China

Corresponding author: 
wxdh100@163.com; 
392292400@qq.com; 
changliang4201@aliyun.com; 
duansh63@163.com

Abstract. The use of customizable or combined modular software products to meet the growing personalization needs of users has become one of the major development directions of the software market in the future. Based on the modular design idea, a highly open CAE software architecture for HAJIF software system is proposed, and a special modular design method is clarified. According to the analysis process of aircraft structural strength, a spatial surface interpolation method based on RBF is proposed for the problem of distributed load transfer to finite element load on space curved structure. The problem is solved well and modularized. The modular design of CAE software is clear, and it can design the CAE system comprehensively, improve the development environment, shorten the development cycle, and have important significance for accelerating CAE software customization.

1. Introduction

With the rapid development of information technology, the use of finite element analysis technology for strength analysis and assessment has become one of the standard processes of aircraft structural design process. Among them, CAE software plays a very important role in the finite element analysis process as a powerful auxiliary analysis tool. In recent years, due to the launch of large-scale domestic aircraft and the replacement of various new types of aircraft, designers have become more and more high-tech tactical indicators for aircraft structures, and the finite element analysis technology in aircraft design has been continuously improved. The traditional load model analysis gradually transitions to the refined model analysis, and the analysis methods and means also undergo great changes. In the aircraft structure design process, in addition to the mainstream large-scale commercial general CAE software system, some host scientific research institutes have more urgent requirements for special feature modules, such as pneumatic load handling, test evaluation analysis, strength check, and engineering database. Therefore, the use of customizable or configurable software modular products to meet the growing personal needs of software products will be one of the major development directions for software companies in the future. Although the commercial software provides the secondary development function, due to the blockade of the underlying core code, the software function expansion is insufficient, and it is difficult to meet the actual needs of the project. In order to better design and analyze the aircraft structure, it is necessary to integrate these special tool modules and technologies into the CAE system software to achieve a high degree of uniformity in the CAE software analysis process.
With the occurrence of the ZTE incident, the domestic intellectual property rights of independent software are becoming more and more important, and the demand and development of special software are becoming more and more rapid. The Aviation Structure Analysis System (HAJIF), as an aerospace industry strength, has developed a civil aviation strength analysis and optimization system with independent intellectual property rights in the domestic CAE field. In recent years, with the development of computer hardware and software technology and researchers for many years throughout the research, HAJIF computing system has greatly improved the solution of scale efficiency [1,2,3] and image visualization [4], and further established its position in the domestic CAE software industry. Therefore, based on the open software architecture of HAJIF system, this paper adopts component development technology and modular design idea in integrated development environment [5], and completes the implementation of pneumatic load processing module and integration with structural strength general analysis platform to better serve. It is analyzed and analyzed in aircraft structure, and it is convenient to provide users with customized development and service.

2. Open software architecture
Based on the modular design idea, with the goal of open, easy to expand and easy to maintain, the HAJIF software system interface, the underlying knowledge resources, the data communication and the external interface (the expansion of new modules and new algorithms) are layered and designed. A highly open software architecture as shown in Figure 1.

![Open software architecture of the HAJIF system](image)

**Figure 1.** Open software architecture of the HAJIF system

The HAJIF software architecture is logically divided into four layers from top to bottom, namely the functional application layer, the execution control layer, the functional module layer and the resource knowledge layer.

- **Function application layer**: It includes graphic visualization function, general function, advanced function and special function, which is an application function that the user intuitively feels.
- **Execution control layer**: Contains the execution control system and data management system. The execution control system is composed of a multi-thread graphics control system and an analysis execution control system, and data communication and storage between the modules are managed by the data management system.
Functional module layer: It contains functional modules such as numerical algebra module, parallel computing module, optimization design module, aero module and special module. It can solve the most basic and single mechanics and mathematics problems and consists of the basic resources of the resource layer.

Resource knowledge layer: It contains the underlying support database such as graphic kernel library, material library, model library, basic common subroutine library, test database and strength criterion library, and provides interfaces for upper layer calls.

Compared with the traditional CAE software architecture, the most prominent feature of HAJIF’s software architecture is its high degree of openness. It has a variety of open interfaces, including database interfaces, algorithm interfaces, dedicated module interfaces, process control interfaces and dedicated development interfaces. Data exchange with the outside world, program addition, and new software extensions.

3. System function modular design ideas

In view of the openness of the framework, the whole software programming consists of a main program and multiple basic service components. The entire build process follows object-oriented and modular software construction techniques, and forms a series of basic service components through dynamic link libraries. Different components work together by defining interfaces to complete data transfer [4] as shown in Figure 2.

Figure 2. Component-based software programming method

3.1 Development environment

The development of the software system is mainly developed under the framework of Visual Studio 2010 using object-oriented programming ideas, specifically using the following tools:

- Computer operating system: Windows 7 64-bit system;
- Programming languages: C++, Qt, OpenGL, Python;
- Programming development platform: Visual Studio 2010.

Among them, Qt is a cross-platform C++ graphical user interface application framework. Because of its good cross-platform performance and interactive communication mechanism, it has become the main tool for data interaction between functional modules and system platforms [6].

3.2 Modular design process

In terms of software coding, the coding tools used vary from the functional requirements of each module, but the integration of each module and the basic platform can be achieved through a unified data interface. The specific implementation process as shown in Figure 3.

The general solver module is mainly implemented by Fortran and C language programming, the
purpose is to make the efficiency of the solution algorithm and facilitate the management of software and hardware;

The front and rear modules are mainly in the Visual Studio 2010 environment, using C++, Qt and OpenGL mixed programming to achieve graphical interaction and display functions [4];

The solver for the dedicated module is mostly implemented by Python script file, compiling the dynamic library or executing program that can be called.

Based on the modular idea, on the software front and rear platform, the interactive Qt graphics framework is adopted to give full play to its superiority in data communication and interface linking, complete UI design and data interaction, and realize its module special performance [7].

![Software modular design development process](image)

**Figure 3.** Software modular design development process

The HAJIF system now has a mature expansion interface and a basic general platform, and its functions are perfect, especially in the integration of new solving algorithms. The new functional modules have the inherent advantages of source code, so deep integration of algorithms and functions can be achieved. Realize the function of the algorithm and ensure the stability of the system performance. The following is verified by the development of the specific aerodynamic load processing module.

4. application examples

According to the idea of modular software development, in the process of strength analysis, the problem of equivalent transformation of aerodynamic load to finite element model load, through module interface development, integrates new algorithm, solves new problems, and expands the new function of HAJIF system.

4.1 Background knowledge

In order to analyze the high-speed aircraft affected by the aerodynamic load, the aerodynamic load and inertial load of the aircraft structure need to be effectively and reasonably applied to the finite element model for analysis [8]. Most of the radomes are hyperbolic structures with large curvature, which cannot be processed by two-dimensional problems. The traditional three-point method, four-point method and multi-point method cannot be used [9]. In this paper, by summarizing the technical characteristics of different load processing methods, the radial basis function has the advantages of high efficiency, convenient storage and simple operation in dealing with multivariate problems [10], and a local radial basis function (Radial Basis Function) is proposed. The load handling method of RBF) solves the load handling of special shape structures such as the fuselage radome. Here, through the software modular design idea, the new algorithm is integrated into the HAJIF software to better serve the structural strength analysis process.
4.2 Surface structure distribution load to finite element load conversion method

The method described here is also used for the airfoil structure. The radome or canopy load is typically provided as a distributed load in a set of spatial points and the lift coefficient thereon, i.e. a set of \( \{(x_i, y_i, z_i), f_i\}_{i=1}^{N} \), intended to be applied to the finite element model. For this problem, we use the method of radial basis function.

The radial basis function was originally applied to the approximation theory and is considered to be the most accurate and stable method for solving discrete data interpolation [10]. Radial basis function interpolation is defined as: for a given multivariate scattered data \( \{X_i, F_i\}_{i=1}^{N} \in R^n \times R \), the radial function \( \Phi: R^n \rightarrow R \) is selected, and use the translation \( \{\Phi\|X - X_i\|\}_{i=1}^{N} \), where the length of the vector \( V \) is expressed \( \|V\| = \sqrt{x^2 + y^2 + z^2} \), and the interpolation function \( S(X) \) is found as determined by the equation (1).

\[
S(X) = p(X) + \sum_{i=1}^{N} \lambda_i \Phi\|X - X_i\|
\]

(1)

Where

\[
P(X) = c_1 x + c_2 y + c_3 z + c_4
\]

(2)

The interpolation condition is satisfied as by the equation (3).

\[
S(X_i) = f_i
\]

(3)

And the orthogonal conditions as determined by the equation (4).

\[
\sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \lambda_i x_i = \sum_{i=1}^{N} \lambda_i y_i = \sum_{i=1}^{N} \lambda_i z_i = 0
\]

(4)

Solving the following equations (5) yields the coefficients \( \lambda_i \) and \( c_1, c_2, c_3, c_4 \).

\[
\begin{bmatrix}
\phi\|X_1 - X_1\| & \phi\|X_1 - X_2\| & \ldots & \phi\|X_1 - X_N\| \\
\phi\|X_2 - X_1\| & \phi\|X_2 - X_2\| & \ldots & \phi\|X_2 - X_N\| \\
\ldots & \ldots & \ldots & \ldots \\
\phi\|X_N - X_1\| & \phi\|X_N - X_2\| & \ldots & \phi\|X_N - X_N\|
\end{bmatrix}
\begin{bmatrix}
\lambda_1 \\
\lambda_2 \\
\ldots \\
\lambda_N
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
\ldots \\
f_N
\end{bmatrix}
\]

(5)

According to actual needs, we use the basis function \( \phi(r) = r^3 \). Substituting the obtained coefficients \( \lambda_i \) and \( c_1, c_2, c_3, c_4 \) into the equation (1), the radial basis function \( S(X) \) is obtained, and all the required values can be obtained by using \( S(X) \). For the convenience of description, the formula (5) is abbreviated as formula (6):

\[
AX = F
\]

(6)

According to most of the papers on radial basis functions, the basis function \( \phi(r) \) is chosen appropriately. The coefficient matrix \( A \) is symmetric positive definite, but in fact this statement does not hold. For example, all points are in the same plane, and the determinant of \( A \) is near zero. The number is large; for example, \( z_i \) is all zero, one row and one column of matrix \( A \) is zero, and the determinant of \( A \) is zero. To this end we use a pseudo inverse matrix or moore-Penrose inverse to overcome this problem.

The matrix \( G \) satisfying the equation (7) is a pseudo inverse matrix of \( A \), and is represented by the symbol \( A^+ \).

\[
AGA = A, GAG = G, (AG)^T = AG, (GA)^T = GA
\]

(7)

Using the singular value decomposition algorithm (SVD), \( A^+ \) is obtained and \( X = A^+F \) is obtained.
The implementation process of the RBF program is as follows:

1) Construct a linear equation formula (5) using the coordinates of the aerodynamic load distribution point and the lift coefficient;

2) Solve the linear equations (5). If the determinant of $A$ is not zero, solve it directly. Otherwise, use the pseudo inverse matrix to obtain the coefficients $\lambda_i$ and $c_1$, $c_2$, $c_3$, $c_4$.

3) The coefficients $\lambda_i$ and $c_1$, $c_2$, $c_3$, $c_4$ are substituted into equation (1) to obtain the radial basis function $S(X)$;

4) For all finite element cycles that need to handle the load, use the radial basis function $S(X)$ to obtain the load distribution lift coefficient of the element node or centroid point;

5) End the aerodynamic load point cycle and output the load card required for structural analysis.

4.3 Software Implementation

Based on the software modular design development process, the implementation of the pneumatic load processing module corresponds to the functional module layer in Figure 1, according to the development process as shown in Figure 3, the implementation process of the pneumatic load processing module is shown in Figure 4.

**Figure 4. Pneumatic load processing module implementation process**

Step 1, use the Python script file to implement the RBF processing algorithm in Section 4.2, and compile and generate the executable file rbf_load.exe;

Step 2, use the Qt graphics framework to complete the interface organization and design of the aerodynamic load processing module in the software integrated development environment (Visual Studio 2010), including the input of the load file. Selection of equivalent methods, equivalent direction, and selection of target objects;

Step 3, based on the modular design, uses the QT communication mechanism to open the input interface of the pneumatic load processing module through signal transmission. The program is implemented as follows:

```python
Connect(actAerodynamic, SIGNAL(triggered()), this, SLOT(activePneumaticLoadMenu()));
```

Through the component calling technology, the pneumatic load processing module inputs the information, starts the rbf_load.exe through the data engine, passes the input data, completes the data connection between the module and the system platform, and the program is realized as follows:

```python
QProcess myProcess; (create process)
myProcess.start(exeFile,param); (Submit input parameters and execute the launcher)
myProcess.waitForFinished(); (wait for the process to abort)
```

Step 4, after the pneumatic load module is processed, the data interface is defined according to the file input format of the HAJIF pre-module, and the real-time display of the load processing can be realized as shown in Figure 5.

The function module can be operated independently or integrated with the analysis platform, which makes the software modules reusable, independent and combinable. It has been used for load handling of multiple models.
5. Conclusion
Firstly, by introducing modular design ideas, this paper proposes a highly open software architecture of HAJIF software system, which facilitates the seamless integration of new algorithms and special function modules, improves the software system functions, and improves the software integration efficiency. Secondly, combined with engineering application, a special method of aerodynamic load based on radial basis function (RBF) is studied for the special hyperboloid structure of radome, and through the modular design idea, from the function input, output and platform link The feasibility division is carried out, and the algorithm is successfully integrated into the HAJIF software system. The implementation of the dedicated module is completed, which fully verifies the important value of the modular design in the design and development process of the complex CAE software system.

References
[1] Chang Liang, Duan Shihui, Wang Likai et al. Large-scale parallel analysis and optimization application of aviation structure[J]. Computer Aided Engineering. 2017, 26(3): 45-50.
[2] Luo Lilong, Chang Liang, Wang Likai et al. Research on the solution technique of hyperscale variable structure sensitivity[J]. Mechanical Science and Technology. 2017, 36(5):816-820.
[3] Wang Likai, Guo Yuchao, Luo Lilong. Implementation of large-scale optimization design in HAJIF system[J]. Engineering and Experimentation. 2016, 56(3): 91-93.
[4] Wang Xiaohui, Huang He, Guo Yuchao et al. Design and implementation of pre- and post-processing modules based on HAJIF[J]. Computer Engineering and Design. 2017, 45(3): 558-564.
[5] Rao Tiangui, Li Kai, Li Yufeng. Modular design of locomotive display application software [J]. Locomotive electric drive.2016, 1:71-74.
[6] Jing Yazhi. Research on architecture design in software development [J]. Technology Guide. 2014, 21: 142-143.
[7] Lu Wenzhou. Qt5 Development and Examples [M], Second Edition. Beijing: Publishing House of Electronics Industry, 2015.
[8] Lei Li, Han Qing, Zhong Xiaoping. Comparative Analysis of the Load Transfer Method of Airfoil Finite Element Nodes[J]. Progress in Aeronautical Engineering, 2014, 3(5): 383-391.
[9] Jin Jinli. Chen Quanli et al. Research on aerodynamic load treatment method [J]. Science and Technology Innovation and Application, 2015, 26: 44.
[10] Hu Chaobin et al. Research on Local Radial Basis Function Interpolation Algorithm[J]. Mechanical & Electrical Technology, 2014,2:43-45.