Send Orders for Reprints to reprints@benthamscience.ae

Open Access

Research on Construction of Web Computing Platform Based on FORTRAN Components

Li Jian[1,*] and Li Xiang[2]

[1]*Jiangsu Vocational College of Finance and Economics, Jiangsu Province Huai An City 223003, China*
[2]*Software College of East China Institute of Technology, Jiangxi province Nanchang City 330000, China*

Abstract: Component-based application service platform by mixed-language programming and loose data coupling mechanism can fully embody characteristics of data storage, data processing and data display. Meanwhile the platform has a good opening performance, and it also contains the mainstream directions of developing the Web application service platform. Combined with the project's developing practice, this paper conducts a systematic study on design ideas of architecture component design, and data coupling of the Web application services platform based on FORTRAN core computing components. Meanwhile, it discusses in detail the achieving ideas of key technologies of components compilation and data coupling, which has a good practical value and certain reference to similar developments.

Keywords: Application platform, components, dynamic link library (DLL), web computing.

1. INTRODUCTION

Web computing based on Web browser as a new web computing mode, is an extension of the distributed computing, Its appearance will eventually extend the distributed computing to the Internet, that has become the basis of modern Web computing [1, 2]. Thousands of personal computers are connected to communicate through it and thus have an inexpensive computing. However, in the process of life in large softwares, new demands will inevitably continue to emerge, and the impact of some demands will be very large for the original system. How to solve some general problems in the large software designing and developing process is always an emphasis and difficulty in the system design. In order to solve these problems come across in the design of a system, one must adopt layered software design strategies, and move gradually from designing the components of the transition to services. Component Services is a new development based on soft Component Object Model (COM) and the Transaction Serve-r (MTS). The component services can deal with many of the resources management original tasks and must be handled by the developers through programming. For example, in the case of the thread security and distribution, by providing thread pooling, object pooling, and run-time object activation, it can automatically provide an application with a greater scalability. COM+ also provides support for data transaction, even cross multiple databases transaction in the network, to ensure data integrity.

FORTRAN was first launched in 1957 by the IBM Company, for the purpose of scientific computing. The initial design took into account the scientific computing and made some optimization for a specific mathematical problem by providing a clear concept of the solution process which is then easily translated into FORTRAN language [3-5]. Comparison of actual operating efficiency also indicates that no matter domestic or foreign, the machine is a classic serial or parallel vector machine, and a lot of experience has shown that in performing the same tasks of scientific computing, FORTRAN is the highest running speed then any other [6]. Therefore, to take better advantage of FORTRAN scientific computing, it has been combined with the current popular Web computing services platform to provide more extensive and efficient Web computing; the idea has a very real sense. In this paper, along with the actual project, FORTRAN computational advantages combined with the current Web service platform are studied, to find a balance between computational optimization and application of maximum; most have certain reference significance [7].

2. WEB COMPUTING SERVICES SYSTEM AND DYNAMIC LINK LIBRARY BASIS

2.1. Web Computing Service System

Based on FORTRAN components, Web computing service application system structure provided by computing the data layer, computing component layer and Web computing application layer and data access interface, component access interface, is composed of a three layer two interface mode.

Computing data layer: data is the basis of computing services. Data is stored by using the database and file mode, The general use of concentrated or clustered patterns stored in the server; Computing component layer: the use of component technology, data computing and data processing functions are encapsulated to form a component library; Web computing application layer: obtain the results of data computing and processing, in an appropriate manner to build an interaction between the user and computer, put the user's information back to computing component layer to begin interactive processing, (see Fig. **1**).

*Address correspondence to this author at the Jiangsu Vocational College of Finance and Economics, Jiangsu Province Huai An City 223003, China; Tel: +86 159 5038 7605; E-mail: charlie273@163.com

1874-4443/15

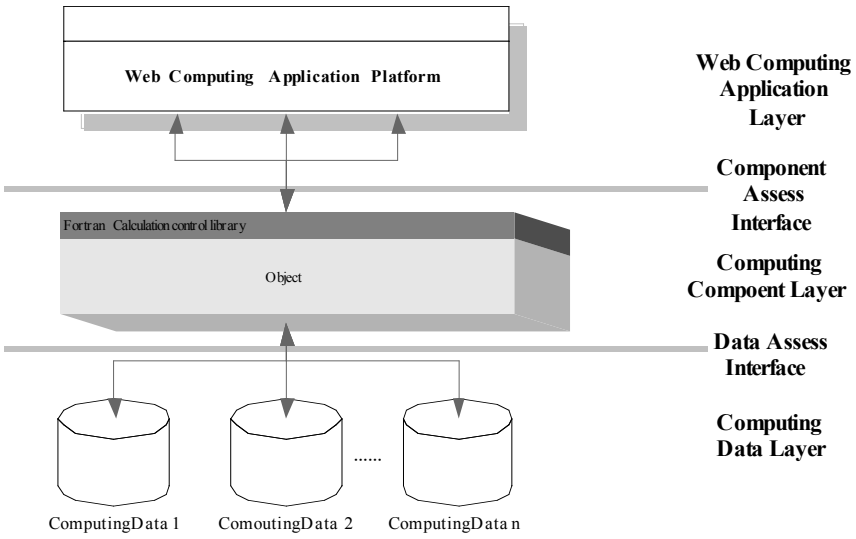

Fig. (1). Three layers two interface web computing service system structure diagram.

Platform data processing flow as shown below:

1) User apply for computing to Web computing application layer by B/S Mode;
2) Web computing application layer standardize the data and pass to computing controller library through the component access interface;
3) Computing component layer perform function based action on data computing or processing for functional requirements, through data access interface to extract data from the computing data layer, which then use their own data processing functions to complete the calculation and processing of data;
4) Component layer sends feedback of the calculation results to web computing application layer, through some suitable ways for showing to the user.

2.2. Dynamic Link Library Basis (DLL)

When create a DLL, you can optionally specify the entry point function. When a process or thread attach themselves to the DLL or separate from the DLL, it calls the entry point function [8]. Use the entry point function to either initialize the data structure or destroy data structure according to the needs of DLL. In addition, if the application is multi-threaded, you can use thread local storage (TLS) at the entry point function to assign each thread a dedicated memory. The following code is an example of DLL entry point function:

```
BOOL APIENTRY DllMain(
    HANDLE hModule, // Handle to DLL module
DWORD ul_reason_for_call, //Reason for calling function
    LPVOID lpReserved ) // Reserved
{
        switch (ul_reason_for_call )
        {
        case DLL_PROCESS_ATTACHED: // A process is loading the DLL.
         break;
        case DLL_THREAD_ATTACHED: // A process is creating a new thread.
        break;
        case DLL_THREAD_DETACH: // A thread exits normally.
        break;
        case DLL_PROCESS_DETACH: // A process unloads the DLL.
        break;
        }
       return TRUE;
    }
```

When the entry point function returns to a FALSE value, if you are using load-time dynamic link, the application will not start. If you are using run-time dynamic link, very few DLL will not load. The entry point function should only perform simple initialization tasks and should not call any other DLL loading functions or termination functions. For example, in the entrance point function, it shall not directly or indirectly call the LoadLibrary function or LoadLibraryEx function. In addition, the FreeLibrary function should not be called in the process of termination. Note: In a multi-threaded application, make sure to use synchronized (thread safe) DLL global data to avoid possible data corruption. To do this, use TLS to provide a unique data for each thread.

To export DLL functions, you can add the function key words to the exported the DLL functions, or you can create a module definition file (.def) to list the exported DLL functions.

Method one: add the function key to the exported DLL function. The function must be declared using the following keywords to export: __declspec (dllexport); the exported DLL function is derived to use in the application, you must use the following keyword to declare to import: __declspec (dllimport). The case is usually, you might want to use a defined statement and the ifdef statement contains header files, in order to separate export and import statements.

Method two: Create a module definition file (.Def) DLL function declaration export. In the module definition file, you can LIBRARY statement and the EXPORTS statement DLL.

The following code is a definition file example:

```
// SampleDLL.def
        LIBRARY "sampleDLL"
EXPORTS
```

3. COMPUTING MODULE DESIGN

3.1. The Process Design of the Module

According to web computing services framework design, the computing capabilities modular, the FORTRAN module, in the background is professionally responsible for the various modules of data calculation and processing functions, through the input/output interface to interact with the front desk users [9]. Modular process design is shown in Fig. (**2**).

3.2. Interface Design

In executing the specific projects, it is carried through the interface function module of data exchange between functions. According to interface function definitions written by FORTRAN 2003 ISO_C_BANGDING, the parameters of FORTRAN sub processes include String that should be set as an array with single character, keep compatible with C, and pass the array length (the length of the string).

Function mult (srcPath, srcLen, descPath, descLen, remark) bind(C, name='foomult')

Integer, value :: srcLen,descLen

Character, dimension(srcLen):: srcPath

Character, dimension(descLen)::descPath

integer::mult

Mult = a * b

End function

The interface parameters as follows:

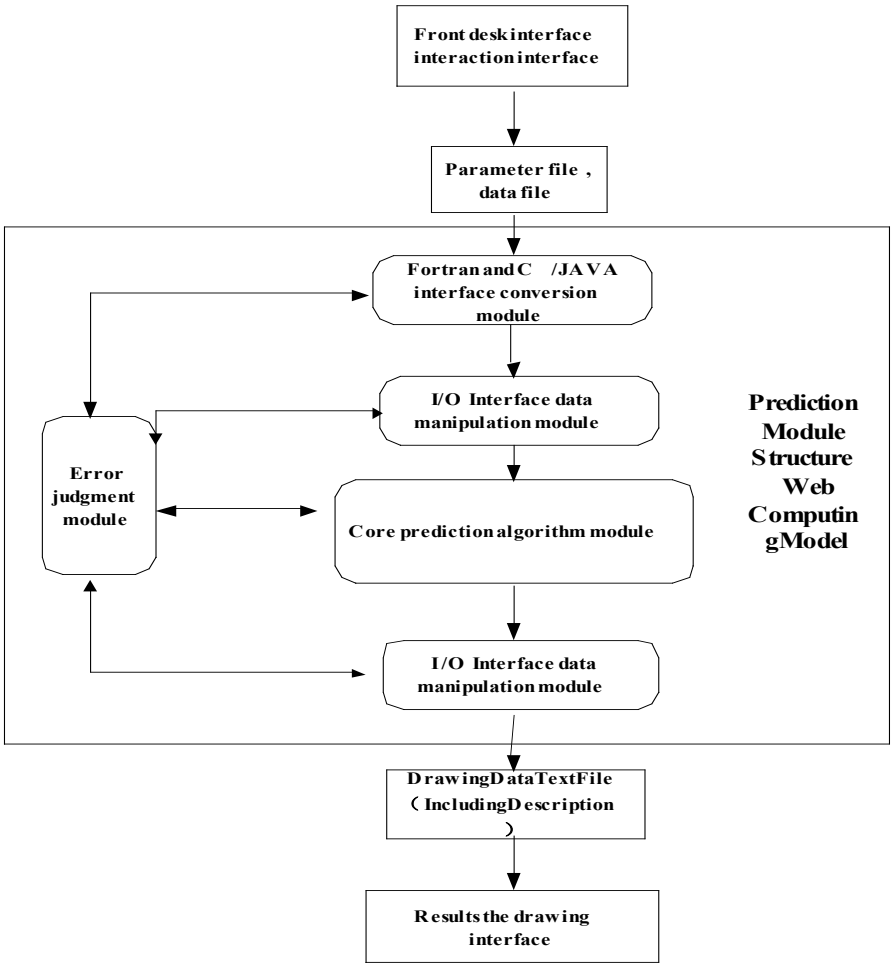

Fig. (2). Forecast computing model of the web module.

1) Interface includes five arguments;

2) The first and the second parameters of the interface used to solve the problem of the input file are as follows: the first parameter contains the path to the file name, such as: /path/in.txt; the second parameter is the length of the first argument. In the first parameter file/path/in. txt the content is organized as follows:

/ path/parameter file1. Txt

/ path/parameter file2. Txt

…… (According to the needs of their respective definitions)

The path one of the input data

The path two of the input data

…… (According to the needs of their respective definitions)

The path of the output data

3) The third and the fourth parameters for output data files, are as follows: the third parameter *i.e.* parameter passing contains the path to the file name, and file naming rules are as follows: user name date/time.txt; the fourth parameter defines the length of the third parameter. For example: the third argument is:/path/zhangsan20101020121120.txt, if the file is empty, do not write specific content, such as data file path and name of the module programmer to write to output. The specific method is: the output data file name is written from the first parameter file in.txt to retrieve the path of the output data, from the third parameter file name zhangsan20101020121120, then add the suffix, according to the nature of their respective file from output contains the path to the output file [10]. For example: (an example for the output data of suffix for. grd,)

/path of the output data/zhangsan20101020121120.grd

If there are data output in the algorithm, write data in file "/output data path/zhangsan20101020121120.grd", at the same time, write the string information in "/output data path/zhangsan20101020121120.grd" to the third parameter file. If there are multiple files generated in the output data, the file naming can be as below:

/output data path/zhangsan20101020121120_001. grd

/output data path/zhangsan20101020121120_002. grd

……(according to the needs of their respective definition)

The generated data file is written above; at the same time write the above string into the third argument to the corresponding file.

4) interface return parameter type integer.

5) BIND (C, name = 'foomult') is another language called the algorithm of dynamic link library name (only change name to foomult in this algorithm, the other remains the same).

3.3. Computing Module Calls

Java/c program can complete FORTRAN calculation module DLL through interface program, for example:

```
#include <stdio.h>
#include <string>
Int main (int argc, char **argv)
{
Char infile[200]=" fszhq_parameter.txt";//Input Parameter File
Int inlen,outlen,remark;
Char outfile[200]="guest20110324.txt";//Output parameter file
Remark=0;
fszhqpredict (infile, strlen(infile),outfile,strlen(outfile),remark);//Call Calculation Module
Return 0;
}
```

CONCLUSION

FORTRAN, as a powerful computing language, has extensive application in the field of scientific computing. But the shortage of language in the field of data interface has greatly restricted its application. Through the method based on hierarchical components design, full use of the FORTRAN is possible, loose coupling way to better handle the relationship besides using other languages [11, 12]. This article provides a FORTRAN component type hybrid application programming summary, and gives a basic introduction on the construction of system and the procedure for web computing platform based on FORTRAN. The components platform is also described in detail with reference to the dynamic link library generation mechanism, CGI interface processing mechanism, and layered interface processing. The study focus also includes summary of FORTRAN multi-language programming to deal with more company problems, in order to produce certain reference functions to similar development.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the Education Science fund of the Education Department of Jiangsu, China (No. JH10-42).

REFERENCES

[1] Bi. Su-ping, Z. Jun, Z. and Zhen-Hong "CVF support for creating a FORTRAN COM components", *Journal of Zhengzhou University (Engineering Science)*, vol. 30, no. 2, pp. 88–94, 2009,

[2] G. Youlin, Y. Wei-ning, and Q. Yan-li, "Based on the VC mixed with FORTRAN programming components", *Microcomputer Information*, vol. 15, pp. 86-89, 2008.

[3] Z. Zhen-Hong, R. Hui, and D. Li-ping, "FORTRAN DLL components are integrated into the NET Framework", *Journal of Wuhan University* (*Engineering Science*), vol. 4, no. 1, pp. 100-103, 2005.

[4] Z.-H. Zhou, D.-R. Zhou, and G.-Lu Yang, "Software component based on COM", *Computer Application*, vol. 3, no. 1, pp. 6-8, 2001.

[5] L. Xiang, "*Research on the Key Technology of WEB Middleware for Mobile Computing"*, Xi'an University of Electronic Science and Technology of China, Information and Communication Engineering, China, 2013

[6] F. J. Feng, and Z.-X. Li, Design and implementation of password cracking system based on Web computing, In: "*National Network and Digital Content Security Annual Conference Proceedings",* 2012.

[7] H.-X. Wang, D.-W. Kang, and X.-F. Wang, "FORTRAN language and origin software and computational physics teaching", *Journal of Xinxiang University (Natural Science Edition)*, vol. 29, no. 1, pp. 87-89, 2012.

[8] X.-T. Li, "Research on engineering calculations and visualization based on FORTRAN ", *Journal of Lanzhou Jiaotong University*, vol. 30, no.1, pp. 98-100, 2011.

[9] H.-J. Ni, and P. Zeng, "Research on the key technology of WEB middleware for mobile computing", *The Information System Engineering*, vol. 7, no. 1, pp. 17-18, 2014.

[10] Y.-M. Zhang, and S.-C. Zhang, "Exploration on FORTRAN teaching adopting numerical calculation program", *Modern Computer*, vol. 8, no. 2, pp. 42-44, 2012

[11] X. Liu and L.-K. Zhao, "Research on distributed remote education system based on Web", *Journal of Shanxi Finance and Economics University*, vol. 33 no. 4, pp. 104-106, 2011.

[12] T. Wu and Y. Li, "Automatic identification processing middleware based on discrete process tracking", *Computer Engineering*, vol. 37, no. 19, pp. 264-269, 2011.

Received: July 18, 2015 Revised: August 22, 2015 Accepted: September 11, 2015