International Journal of Hybrid Information Technology
Vol. 7, No. 6 (2014), pp. 433-444
http://dx.doi.org/10.14257/ijhit.2014.7.6.37

The Research of Software Reuse Technology Based on Component

Zhai Jinbiao

State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, BeiHang University
jinbiaozhai@gmail.com

Abstract

With the expansion of the computer application, the complexity of software and the program code increase rapidly. The increasing requirement of software brings the grown of the size about software. In order to solve these problems, the idea of software reuse emerges. In these years, the software reuse technology develops rapidly. Nowadays, software reuse technology which is based on component is regarded as the most successful implement technology in the software reuse field and the available solution to solve the Software Crisis. In this paper, we research the software architecture based on component. And we propose a model of library management information system. Firstly this paper introduces the research background and research status. The second part is the design of reuse system based on the component. The third part is the component-based information management system used for the library management system. The last part is the conclusion.

Keywords: *Software reuse, Component, Software architecture*

1. Introduction

In the traditional economy, the requirement of the enterprises for the software is functional and high quality. Now, enterprises put forward higher requirements for the update speed of software with the diversification of customer demand and the increasing competition. Over the past years, enterprises may update the software every year. But now, enterprises must update their software every half year, every three months or even everyone month. On the other hand, over the past years, the software is mainly used to improve the production and enhance the internal management. Now it becomes a tool for the enterprises to gain greater benefit gradually. It also becomes an indispensable element in people's life and work. The change makes software face a new challenge that is how fast and high quality to develop the different software to meet different demands. The traditional software development methods have no capability to meet the demands. It becomes the best solution for solving the problem to establish a better software development environment, develop and adopt the reusable software components.

In the process of software development and maintenance, we have to face a series of serious problems. Scientists in Western computer circles call these problems as the software crisis. The software crisis appeared firstly in the late 60's. Its main performance includes some aspects: the quality of software is poor and the reliability is unwarrantable; software do not finish the tasks within the budget; it is difficult to control the software development; the cycle of software has to prolong; it is difficult to maintain the software. The maintenance personnel and maintenance costs continue to increase [1, 2]. With the constant development of the distributed object technology, we need an effective technical approach to alleviate the software crisis. The distributed object technology provides strong support for the software reuse [3, 4].

ISSN: 1738-9968 IJHIT

In order to put forward and develop the thought of the software reuse, there is a far-reaching historical significance and practical significance to develop the computer technology [5]. In general, it will get the overall quality assurance to use the reliable component design and the application system. It brings the mark opportunities which can improve the company's economic benefit and the social benefic [6]. In the NATO software engineering conference (the north Atlantic treaty organization) in Gramish in 1968, Dough Meilroy put forward the concept of the software reuse in his paper: Mass Produce Software Component. But after ten years, the research of software reuse has not made the substantial progress [7, 8]. In 1994, International Discussion Conference of the software reuse was renamed the International Conference on software reuse. The technology of the software reuse has attracted wide attention in Computer Science. More and more people begin to study the technology. Caldieri and Basili proposed the software development process based on reuse. IBM and Loral Federl system company developed the reuse maturity model *etc.*, [9, 10]. The technology of the software reuse has applied into many fields in the domestic such as the design of reusable structure for small satellite system [11]. For example, the DDS based on the component technology has obtained satisfied results because of using the thought of the software reuse [12]. With the widely spread of Internet, it provides a compared method for connecting geographically dispersed entities. The Internet has many users. Software reuse is convenient, interactive, and low cost and its functions are powerful. It provides the opportunity for the software component industry [13, 14]. With the development of Internet, the patterns of the software industry have changed. The software industry based on the computer network has changed itself from the closed software development into the development model which is based on the software reuse. In the future, the technology of the software reuse and activities will become more systematic and standardized. There will appear more organization using the reusable technology. There will produce software reuse development and the CASE tools which support the reuse and analysis. It must emerge the domain specific reusable software component factories. It will form the development standards for supporting the development of the reusable software [15].

In this paper, we analyze a software reuse technology and the method of the software development comprehensively. According to the domain analysis results and the theory of knowledge system architecture, we propose a domain software structure which is based on a library information management. The architecture can obtain the functional model of the main components and the hierarchical model of the system through detailed field analysis.

2. The Design of Reuse System based on the Component

Over the past few years, software reuse technology has been integrated into the mainstream of software engineering construction. Many software design technologies (such as visual programming) have become preferred for many engineering technical personnel and system developers. In fact, the real benefit of the development environment which is provided by these techniques and tools is that they have the infrastructure support for the software reuse.

The development process of an application software system usually includes several typical stages. They are required analysis, system design, coding, testing and maintenance. Adopting software reuse technology can avoid the duplication of each application system effectively in each stage. Its purpose is that the application system development is no longer all start modes. It makes full use of the accumulation of the past development experience and knowledge (such as demand analysis, design, test plan and maintenance and so on) which is based on the existing achievements. And it can be used in the source code and function module directly or with a little modification.

The design of software reuse system which is based on the component is similar to the process of the general software engineering design. It is mainly composed by the following several aspects:

(1) The demand Analysis

The users or the system analysts analyze the results and combine with the special requirements of the application system. Then they write the system function and use the informal description which is based on the understanding of the user requirements.

(2) The functional analysis

They describe the system function by using the formal language. It includes the reusable parts and parts which can be used as reusable components. The output is the functional description and the reusable building description. It can be placed in the overall design process.

(3) The Detailed Design

They use the reusable components to build the algorithm design of the software. In this stage, the detailed algorithm will produce the functional description of the reuse components in order to generate a new reusable component.

(4) The Coding

It constructs the executable code. It includes the direct use of reusable components and the new reusable components which are generated by the component interface rules.

(5) The Operation Maintenance

Maintenance is performed at the functional level.

(6) The software reuse and understanding

The software and the design specification which has the full test can be used as the reusable component storage.

In the stage of the detailed design and the coding, the design of software reuse will involve the retrieval, modification and combination of the components. In the final stage, it will also face the question of the component storage. Actually, it is the core development model of the software reuse. It can be described as following:

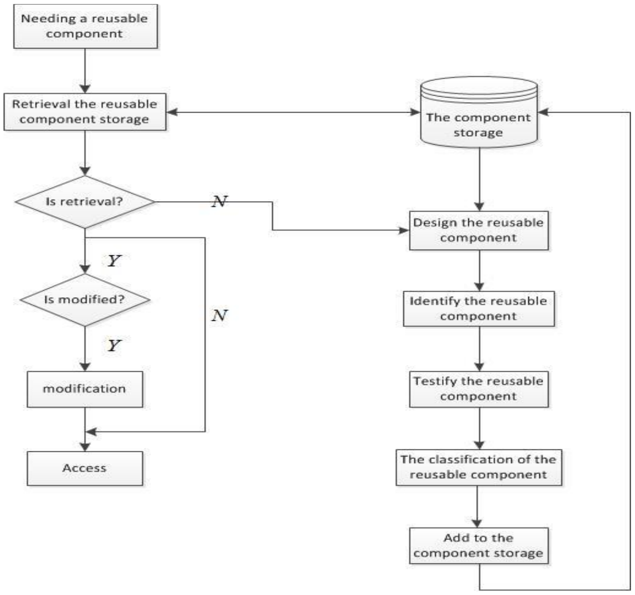
Needing a reusable component
Retrieval the reusable component storage
The component storage
Is retrieval?
N
Design the reusable component
Y
Is modified?
N
Identify the reusable component
Y
modification
Testify the reusable component
Access
The classification of the reusable component
Add to the component storage

Figure 1. The Design Model of the Software Reuse

The prior identification is the prerequisite to achieve the reuse. It can search the reusable components fast. Reliability and correctly are very important for the reusable components. After the reusable components are identified, they must be tested strictly and reviewed, that is, to verify the reusable components. The final process is to classify before storage. It is the declarative description for the function, usage, scope of application and interface of the reusable components to prepare for searching the components.

2.2. The Design Method of the Components

The software reuse technology based on the components is different from the method of various kinds of software construction. It focuses more on the retrieval and the method of the reusable components. It assembles the software system using the existing components. The acquisition of reusable components is the premise of the software reuse system based on the components. Designing reusable components from the application field is the key to implement the system.

The method of the software components is to design and implement a specific function for the component object. Firstly, this method designs the reusable components. And then it combines the component objects to build the software system. The method is based on the object oriented. When this method is implanted, it can be used immediately. This method makes the concept as the center. And it builds the technological system of the application system by the organic combination of the components. The components have some features such as visualization and standardization. So the method is easy to use and can be executed immediately.

The components are based on the object orientation. The introduction of the concept for an object can divide components by objects. The component object is the package of all kinds of features (entity attributes and relevant operation). And it embodies the abstract entities and the relative independence. Packaging hides the component’s internal data and operational details. Components communicate through a certain interface so that the change of the internal implementation of the components does not affect the other parts of the system. Components communicate follows the principle of information hiding. In the inheritance hierarchy mechanism, the low component object can inherit all the features of the upper object (attribute and operation).It is very convenient to increase and expand the components. Unlike traditional component object module calls, the component object links the component objects by adopting the method of the message driven. So the component object can achieve the parallel and distributed processing among the components in the greatest degree.

A component is not for a special software system but for multiple applications to share. Therefore, the process of software reuse must be around the goal to share from the field analysis, designing to the component extraction, description, classification, certification, testing and storage. It requires the design of the components must adhere to the following principles:

(1) It needs to improve the abstract level in order to enhance the reusability of the components to make full use of the inherited characteristics of the components.

(2) It is comprehensible, readable and easy to modify strongly. The component design should have a very good semantic interface. It should also have a document of complete, correct and easy to use in order to facilitate, expand and improve the functions of the components.

(3) The components must have a high cohesion and a very low degree coupling.

(4) The strong comprehension. The components can be not only easily integrated but also have the flexible decomposition according to different application. Then it needs to make the variable components be data and parameter in order to suit different applications.

(5) A strong downward compatible ability. Component storage must have a strong ability of the version control to facilitate the upgrade of the components.

(6) A strong ability of evolution. The data and its structure package together. Data should be stored in the data component object. It can explain the structure initiatively. It is the basis for components to interact.

2.3. The System Structure of the Application Software System based on the Components

2.3.1. The Software Architecture: In the application, software system is based on the components. But the complex control of the system is a very important problem. The first software system structure based on the components is described informally by diagram. Such diagram and description bring ambiguity. The mean is based on the common intuition and the historical experience entirely. More and more people realize that the effective methods of the software system must implement the design and architecture level. The strong software architecture will bring the great efficiency directly for the development and maintenance. It is mainly manifested in the understanding of the system, organization development, encouraging the reuse and the evolution system *etc*. The greatest contribution of the software architecture is to improve software productivity and simplify the maintenance. The key to improve the productivity of the software is reusing parts of the software. The key to simplify the maintenance is reducing the software costs and improve the software quality. It is the purpose to study the software architecture.

The software architecture based on the components is an organization structure as a program or a system constructs. The software architecture is the principles and guidelines to relate and control the system design and development. The basic of the software architecture definition based on the components is the structural properties of the system such as the structure, the relationship of the components and the whole system framework.

The software architecture is used to describe considerably the composition and the structure of the system. It defines the system according to the contact and constraint among the functional components. At present, there are several common system architecture models, such as filters, data organization, layered system and repository heterogeneous architectures etc. The study on the architecture of the application software system based on the components is a new research direction.

2.3.2. The Hierarchical System Structure: The system structure is layered architecture. The component is usually divided into three layers: the basic component layer, the domain general component layer and the domain specific component layer. It is described as following:

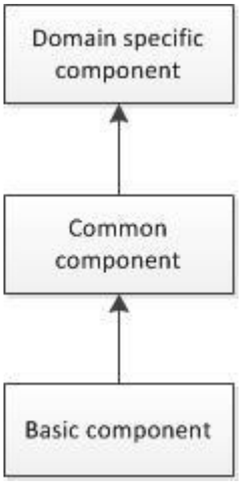

Figure 2. Component Hierarchical System

The bottom is a foundation component. It is the component which is used in the integrated support environment and operational environment. In this paper, it refers to the tool mean, data window, buttons and the other components. The middle layer is a field of universal component such as the report component of MIS and the query component. The top is domain specific component. The construction is lower level, the function is much simpler and its flexibility is higher. The construction is higher level, the function is stronger and its flexibility is poorer.

3. Component-based Information Management System used for Library Management System

3.1. Module Diagram: we analyze the library management system on the basis of the demand analysis for the library management system. And we propose a software architecture based on library management Information Systems: Component-based Information Management System. It is used for Library Management System-CIMSLMS.

The overall design idea is analyzing the library to get architecture through data collection, research, demand analysis and the dormitory management situation. CIMSLMS includes these parts: Maintain, Circulation, Interview, Catalogue, Reading, Archives, Readers management and Statistics. We only analyze the main modules.

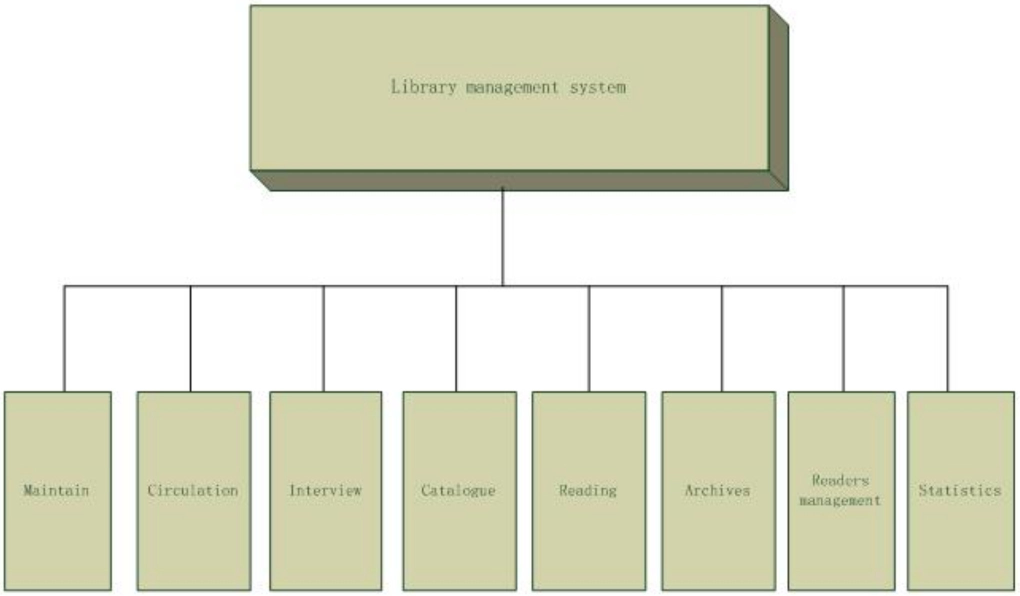

Figure 3. Library Management System Function Module Diagram

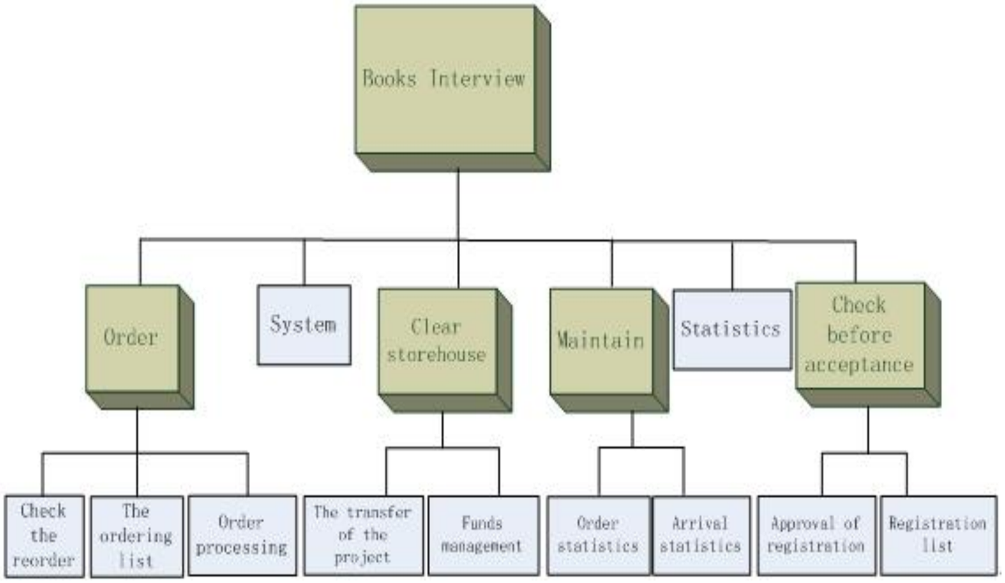

Figure 4. Interview Modular Structure Chart

Interview is the job that staff survey and collect the library information. This job needs to meet the demand of readers through the investigation on the collection, the reader and the book source.

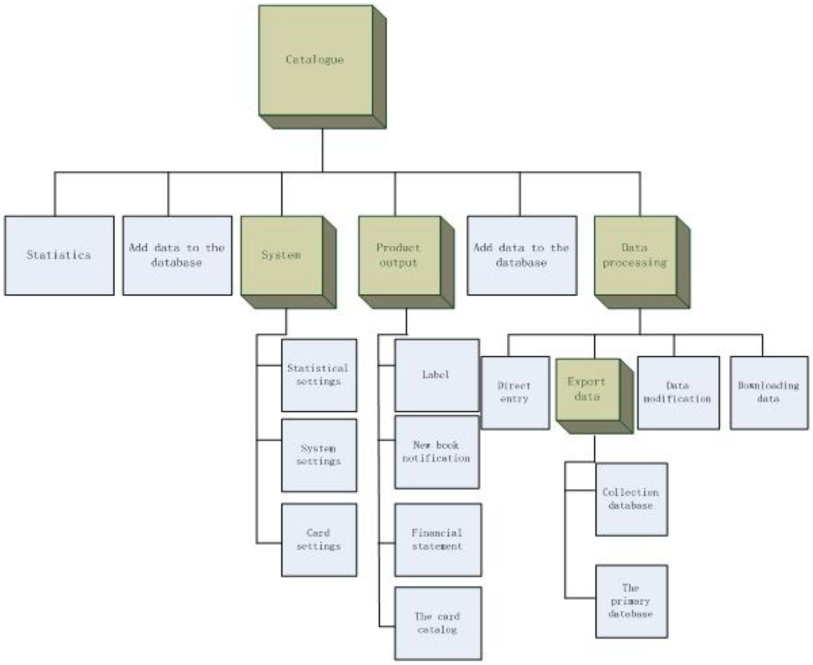

Figure 5. Catalogue Module Structure Diagram

These are many works in library catalogue: (1) New book received (2) Check (3) Description (4) Classification (5) Subject indexing (6) Call number selection (7) Printing bookmark and paste (8) Send to the library

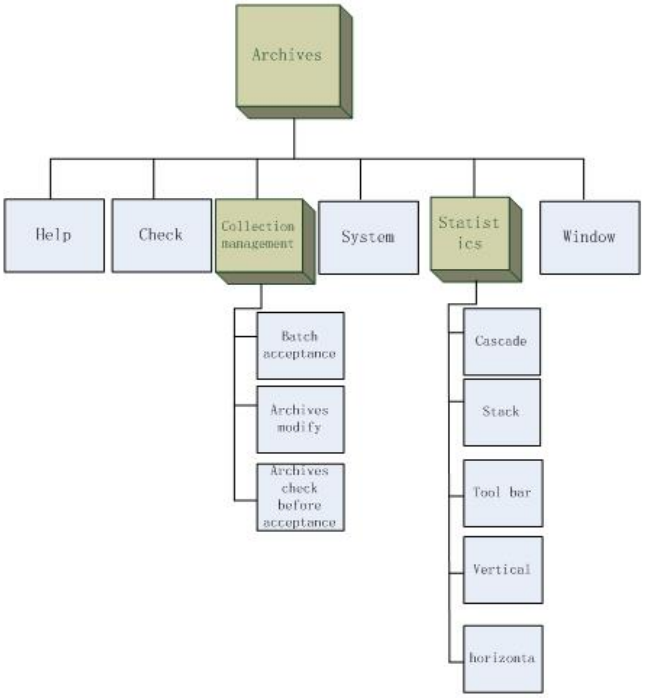

Figure 6. Archives Module Structure Diagram

Archives work is an important part of library work. It mainly refers to the book acceptance, distribution and rational layout.

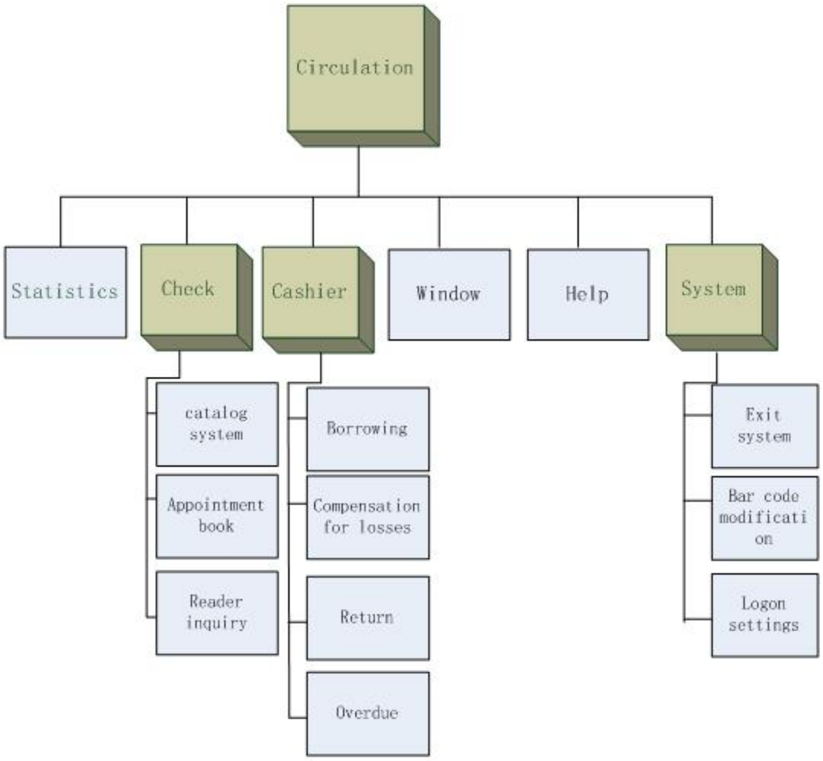

Figure 7. The Circulation Module Diagram

Circulation is responsible for reading service, tape recording, propaganda, reader consulting, handling procedures, preservation and repair.

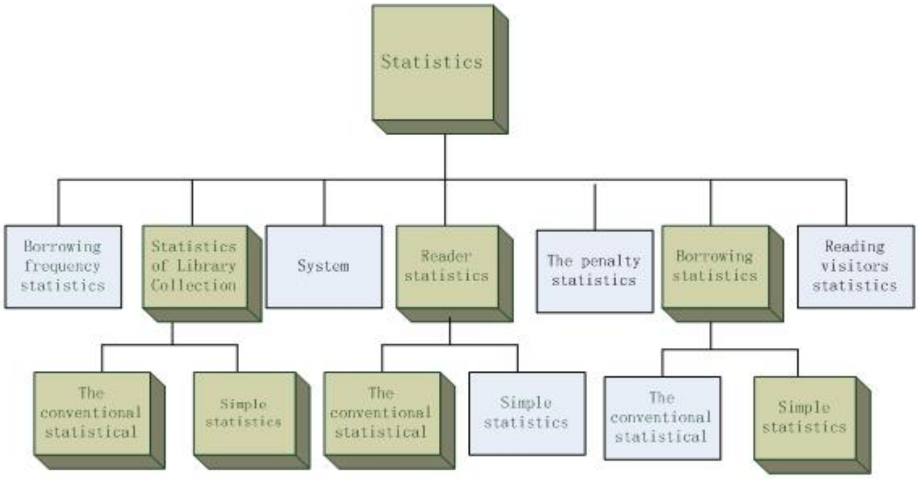
Statistics
Borrowing frequency statistics
Statistics of Library Collection
System
Reader statistics
The penalty statistics
Borrowing statistics
Reading visitors statistics
The conventional statistical
Simple statistics
The conventional statistical
Simple statistics
The conventional statistical
Simple statistics

Figure 8. The Statistical Module Diagram

Statistical function in library management system are required to statistic all book conditions. These conditions include borrowing condition, readers' fine condition, and a report of loss, damage the books. This function provides the print statistical results.

3.2. The Three Layer Structure of CIMSLMS and Development Environment: CIMSLMS system adopts the three layers structure of a computer system. The third layer application software model showed as following:

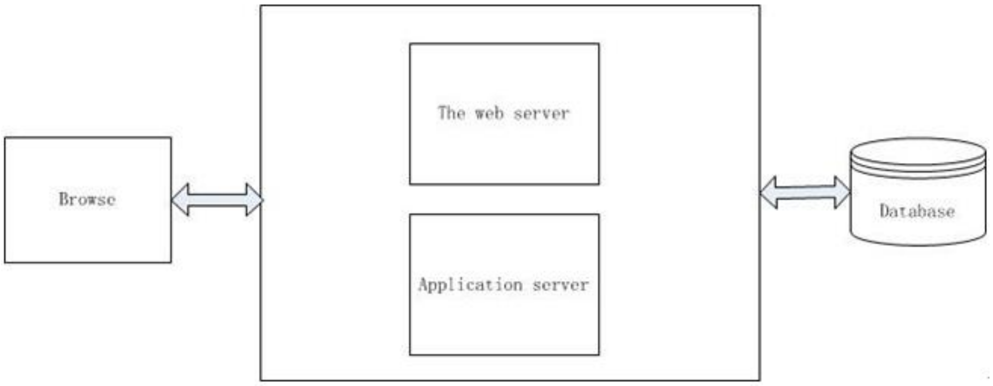

Figure 9. Three Tier Application Software Model

The first layer is service layer. It is responsible for dealing with the interaction of user and the application server. It is also responsible for the implementation of the client system data display and operation, as well as the reasonableness check for users.

The second layer is application layer. It is between the client layer and data service layer. It is the interaction with customer service layer and data service layer. Application layer is responsibility for receiving the client request and transforming the request to database request.

The third layer is data service layer. This layer processes the database operations according to the request the server send. It provides data for the Web server and application server.

3.3. The Architecture of CIMSLMS: CIMSLMS adopts three layers system structures and its structure is showed in Figure 10. The horizontal is three layers of application. The longitudinal is the component level.

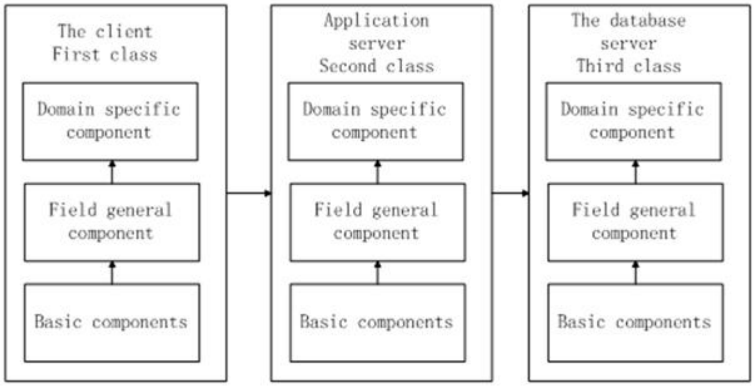

Figure 10. CIMSLMS Component System

From the data point of view, we can see that components can be divided into three categories: the first category is related to the client data, and fits in the client; the second category is related to the client and the background data, and fits into the application server; the third category has a relation with the background data, but it has nothing to do with the client data. It is suitable for on the database server.

4. Conclusion

Software reuse is a solution to avoid repetitive work in software development. We can improve the efficiency and quality of software development by using the software reuse. Research on software reuse has become a hot spot and it is regarded as a feasible way to solve software crisis in reality. Nowadays, software component technology is regarded as a key factor in the field of software reuse technology. We can see that using software reuse technology widely will promote the development of software industry. The software component production will become an independent industry. The development of software systems is bought by system integrators though the purchase of commercial components integrated assembly. The development of the reform will be a good opportunity in Chinese software industry.

According to field knowledge system analysis theory and system structure, we try to combine the library management information system and propose a library management based on software architecture. This model gets the hierarchical system. The main components of the system are based on the detailed analysis of the system structure field. We introduce the system design thought briefly. And we research on hierarchical structure and CIMSLMS system structure based on the components. Then, we analyze the components and the functions of the CIMSLMS. We prove that there are outstanding advantages and strong vitality in the software reuse and reusable component technology

References

[1] W. Frakcs and P. Gandel, “Representing Reusable Software, Information and Software Technology”, vol. 32, no. 10, **(1990)**, pp. 654-664.

[2] W. B. Frakes and B. A. Nejmeh, “An Information System for Software Reuse. In: Software,Reuse: Emerging Technology”, IEEECS Press,**(1990)**, pp. 142-151.

[3] D. Embley, *et. al.*, “OO System Analysis: Is it or isn’t It”, IEEE Software, vol. 12, no. 7, **(1995)**, pp.19-33.

[4] D. A. M. Lessandro, *et al.*, “The Generic Reusable Component: An Approach to Reuse Hierarchical OO Designs. In: Proc Advance in Software Reuse”, Los A lam itos,California: IEEE Computer Soeiety Press, **(1993)**, pp. 39-46.

[5] D. Garlan and D. E. Perry, “Introduction to the Special Issue on Software Architecture”, IEEE Trasactions on Software Engineering, **(1995)**, vol. 21, vol. 4, pp. 269-274.

[6] B. A. Burton, *et al.*, “The Reusable Software Library”, IEEE Software, **(1987)**, pp. 25-33.

[7] NATO Communieations and Information Systems Ageney, “NATO Standard for Management of a Reusable”, Software Component Library,**(1991)**.

[8] NATO, “NATO Standard for Software Reuse Procedures”, NATO contact number CO-5957-ADA, **(1991)**, p. 3.

[9] F. Yang, W. Shao and H. Mei, “Design and implementation of JBIII type object oriented system in CASE environment”, Science in China, **(1995)**.

[10] F. Yang, H. Mei and K. Li, “Software Reuse and Software Component Technology”, Acta Electronica Sinica, **(1999)**, vol. 27, no. 2, pp. 65-75.

[11] B. Hong and D. Lu, “The Reusable Architecture Design Based on Software Bus for Small Satellite On-board System Software”, Computer Engineering, vol. 25, no. 8, **(1999)**, pp. 54-57.

[12] S. Y. Fu, J. Zhang, G. Wang, Z. Xu and J. R. Liu, “Research and Design of A Component-Based Distributed Software Platform”, Mini-Micro System, vol. 20, no. 12, **(1999)**, pp. 885-889.

[13] Acta Electronica Sinica, “Research of Dynamic Data Technology for Intranet”, Acta Electronica Sinica, vol. 27, no. 8, **(1999)**, pp. 99-101.

[14] Y. Sun, W. Zou and M. Liang, "A Component Library Management System Based on Internet", Computer Engineering, vol. 26, no. 11, **(2000)**, pp. 85-86.

[15] Q. R. Li and Z. F. MAI, "Domain Analysis: Producing Useful Models for Software Reuse", Journal of Computer Research &Development, 36, no.10, **(1999)**, pp.1188-1196.

Author

Zhai Jinbiao, he was born in 1983. He received his bachelor's degree from Beihang University. Now, he is a PhD in Beihang University. His main research interests include software reuse, distributed workflow, computer security.