

Method of Software Development Based on Component Technology

Li Yan

Ningbo Dahongying University
Ningbo china 057488052105
zyu4262@163.com

Jiang Li

Ningbo Dahongying University
Ningbo china 057488052105
jiang2000@126.com

Abstract — Component-based software development methodology is a software development reuse software solutions in order to avoid duplication of effort, the current software component technology is regarded as one of the key factors of successful reuse. The article describes the main content of the basic concepts of software reuse and component technology. Based on component-based software development technology in the development process of the actual system are discussed, and the development model are proposed.

Keywords-software reuse; software component technology; software development methods

I. INTRODUCTION

In recent years, component-based software development (CBSD) technology was valued by the growing number of software industry. Granularity improve the software reusable component-based software development is conducive to the improvement of software quality and the efficiency of software development, contribute to software industrialization to achieve.

With the development of software engineering, software reuse gradually becomes a hot topic. Component-based software development methodology has changed the traditional software development methods which based on "data structures + algorithms" development models. This method is able to reuse the existing achievements, and thus be respected and the rapid development of the majority of developers.

II. SOFTWARE REUSE TECHNOLOGY

Software reuse refers to the process of repeat run reusable software which were developed by the purpose of reusing. Reusable software refers to software designed for reusing purposes. More emphasis on software reuse is the purpose of reusing. It is a systematic method, which was designed and developed for reusing. And to effectively organize and manage these complex products easy for people to find and use, and development based on reuse. To effectively organize and manage these complex products, which were easily found and used by people, in order to facilitate the development based on reuse.

According to the multiplexing method, we can software reuse into the black box and white box reuse. Software reuse is divided into black box and white box technology. The black-box reuse on reusable components without any

modification, reuse unchanged. This type of reusable components more common, it tends to have a good package, and has a standard interface. The multiplexing rate of such component tends to be relatively high, thus requiring a very high quality and reliability. White-box is multiplexed with a component shall be partially modified to adapt to the needs of the specific application.

III. SOFTWARE COMPONENT TECHNOLOGY

A. Components and component technology

Components refer to the constituent components of the application system can be clearly recognizable. Reusable components have relatively independent of the function and value of reusable components.

The software component is relatively independent functional software system, can be identifiable interface specified by the contract, there are obvious dependencies and context can be independent deployment can assemble software entities^[2]. By the interface, implementation, and deployment of three elements[3], Which the interface mainly explained parts that can be completed to achieve the code so that the operation of this component, deployment exist in the form of the component, namely the binary code or executable files. Software Component Technology (Software Component Technology) is the core technology to support software reuse, and attaches great importance to the rapid development in recent years and are subject to a disciplinary branch.

B. software develop model based Component

Today's software development using existing resources, composition, customization and reuse in different levels.

- To provide customized means. The application software can be used to pre-programmed feature clear parts custom made and can be different versions of components to achieve the expansion and updating of the application.
- The software component technology. Decomposition of complex systems are difficult to maintain as independent of each other, the components work together and strive to make these components can be used repeatedly.
- To achieve cross-platform interaction. Break through the restrictions of time and space, and different

hardware, unified interface to achieve cross-platform interoperability.

IV. THE BASIC IDEA OF THE METHOD OF SOFTWARE DEVELOPMENT BASED ON COMPONENT TECHNOLOGY

Software will need to design the basic idea of the method of software development based on component technology that is broken down into some of the basic functions of the components do not have to redesign these basic functional modules in the development process, and just finding the right component assembly from an existing component library into the software, the idea of this software industrialization will undoubtedly greatly enhance the software reusability.

Software development based on component technology not only makes the software products in the goodness of fit customer needs, and ahead of similar products in the on-line time, software quality. This technique improves the success rate of the project, and the development and maintenance of software becomes very simple, customers can respond to changes in the business environment and changes in IT technology anywhere in order to achieve rapid customization.

The basic purpose of the software development method based on component technology develops application system based on component assembly. This assembly is based on those components developed in the form of independent service membership. The service of Components is to invoke some common services on certain common infrastructure for information exchange. Therefore, software development based on component technology has the following elements: by application component assembly, independent service providers, public components of the infrastructure and universal service.

V. THE PROCESS OF SOFTWARE DEVELOPMENT BASED ON COMPONENT TECHNOLOGY

Software development process based on component technology, as shown in Figure 1, this process has five main components: needs analysis to establish the corresponding component library, component acquisition, components adjustments or modifications, components assembly.

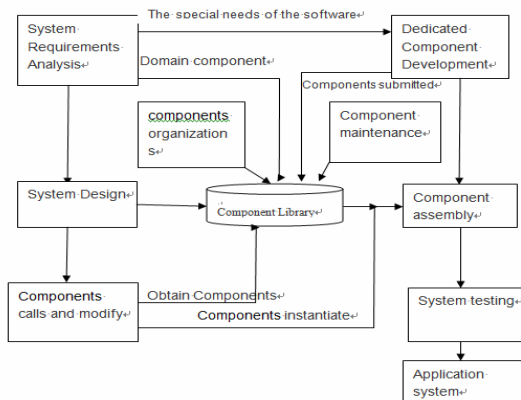


Figure 1 the process of software development based on component technology

A. *analysis requirements, design the overall structure of the framework*

First, we should analyze the entire system requirements and then compare the same field of business needs to identify the common functional requirements of each other. Functional decomposition based on the results of requirements analysis of the target software system, the system is decomposed into strong cohesion, weak coupling of the functional components of component-based development system.

B. *establish the corresponding component library*

We must establish the appropriate component library to meet the development needs of the various levels of software component technology-based software development. Component library is summarized management software component to form a large manageable, searchable, reusable library management. Component library as an infrastructure to support software reuse and software asset management facilities, provides the main service of software component description, classification, storage and retrieval functions, and at the same time to provide effective support for software development based on component technology to improve the efficiency of software development and software product quality.

C. component acquisition

According to the results of requirements analysis and components library, we began to seek components to meeting the functional needs, then assessment and adaptation, and finally assembly to the corresponding functional decomposition framework.

There are three ways to acquire components:

- The member is selected in accordance with the principle of new system development.
- According to the new functional modules in the current development system to the development of new components.
- Suitable components or purchased through commercial use of Internet resources to be shared or free access.

D. components adjust or modify

In an ideal world, the total existing components can be applied without any change, but in most cases, that can not be done. In some cases, the existing reusable components provided not match with the new system architecture or design rules. We need to obtain the components function on the simple adjustments or modifications to minimize the conflict between the components. So that can meet to the needs of the current function.

Common methods include: white box. Users can modify source codes so that components can interact with other components. Black-box, users can get binary executable codes, components do not provide extension mechanism or API. Gray-box, source code can not be modified, but It provides itself extension mechanism member or API.

The assembly of components integrated to form a development system.

Component assemblies interconnected components in the component library after appropriate modifications, or connect them with the software elements in the current development projects, and ultimately constitute a new target software. If the individual components want to play a role, it should integrate with other components, common integrated into a well-defined framework, this architecture can be completely different components combined. Usually the process is binding the atoms component as far as into a larger composite member, and then assembled according to the interface member.

In the assembly of the process of designing a new system, component developers, according to the actual situation of the network and equipment deploy software, such as the versatility components deployed in a central data server, some special components are deployed in the most convenient to the user server, even the user's machine. This allows the components to deploy optimal. In addition, the process of assembling components sometimes have to write glue code, these codes can eliminate the problem of incompatible interfaces between components, as well as the system provides a unified exception handling mechanism.

VI. CONCLUSIONS

With the growing demand of software, software size and complexity is increasing, and people began to explore new

software development techniques to adapt to the requirements of software development. Component technology by creating components, reusing components and assembly components complete the development of software systems. This can improve the efficiency of the system development, and reduce the development costs of the system to meet the requirements of software reuse. Thus, component-based software development methodology is an effective way to solve the software crisis.

REFERENCES

- [1] Yang Fuqing, Hong Mei, Hacken Lee Software Reuse and Software Component Technology [J] Journal, 1999, 27 (2): 68 - 75.
- [2] Yang Fuqing, Hong Mei. Component-based software design and realization [M] Beijing: Tsinghua University Press, 2008.
- [3] XuZhen. Component-based software development methods and realization [J] software GUIDE, 2009 (11) :17-19.
- [4] Li Shuang. Creating component platform to promote software production changes [J] China Awards for Science and Technology, 2007 (5) :22-25.
- [5] Ye Junmin, Chen Zhuo, Lei Zhixiang component assembly applications software development process research [J]. Application Research of Computers, 2008,25 (6) :1736-1738, 1752
- [6] Xu Xuefeng, Fengshu Xing, Chen Shilu component-based software development methodology and technology research [J]. Journal of the Academy of Equipment Command & Technology, 2007,18 (5) :94-98.
- [7] Wang Ya Fei Deng, Gan Wenli reusable component-based software development model [J]. Computer Knowledge and Technology (Academic Exchange), 2007 (20):451-452, 456