Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (8): 1233-1238, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.1233.1238

Study and Application of Legacy System Reengineering based on Component Reuse

[1]Ai-Ping Li, [1]Zheng-Hua Wang, [1]Li-Guo Duan and [2]Xue-Ping Li
[1]College of Computer Science and Technology, Taiyuan, China
[2]Polytechnic Institute, Taiyuan University of Technology, Taiyuan, China

Abstract: This study proposed a solution of hierarchical architecture reused model to realize legacy system reengineering for the secondary development process of endowment insurance based on the third-version core platform of China Social Insurance Management System of golden insurance project. In the model, every legacy function component is wrapped in the unified interface standard of SOA to realize legacy system reengineering application based on component reuse. The development of illustration system and components shows that the solution provides a good mechanism of realization and safeguard for legacy system reengineering and the solution can also provide more beneficial reference and help for legacy system reengineering during software development and integration in cloud computing platform.

Key words: Legacy system reengineering, hierarchical architecture, component reuse

INTRODUCTION

As one of the effective ways of improving the software development efficiency and resolving software crisis (Frakes and Kang, 2005; Liu, 2012), software reuse refers to reusing items like project plan, cost estimation, architecture, demand model and protocol, design, source code, technical document, user interface, data and testing examples and so on. Critical factors in its realization can be classified into technical and nontechnical ones. Technical factors include software component technology (such as the extraction and realization of component and the management of component library), domain engineering, software architecture, software reengineering (such as transforming legacy system and extracting reusable components), CASE tools and so on.

Legacy system reengineering is an important component in software reuse (Hui *et al*., 2010), because it can extract critical operations and reusable components from the legacy system so as to form a strategy to transform it. In cases of network and distribution, there are great market demands for legacy system reengineering.

Recent years have witnessed vigorous development of social insurance businesses. As a result, the core platform of China Social Insurance Management System of golden insurance project (Ximing, 2010) has been revised two times after its first release in 2000 by the former Ministry of Labor Security, one in 2003 and the other in 2009. The third version not only follows and extends previous technical framework, but also makes innovation in business system and data model. Based upon the third-version core platform, local security departments can make secondary development to incorporate local security businesses. This study is aimed to explore the legacy system reengineering based on the component reuse in the secondary development of endowment insurance in Taiyuan social security system.

CURRENT STUDIES

Software reuse is not only a major research domain in software engineering, but also an important orientation to resolving software crisis and developing software industry. In recent decades, object oriented technology and distributed object technology have emerged and gradually become the mainstream, which provides basic technical support for software reuse. The software reuse process has undergone three revolution: from programming code, library functions, program family at the very beginning to domain requirements, software architecture, components, design patterns, test cases and later covering software measurement method, process managements etc. The above reuse process provides systematic, all-around support and improves the credibility and development efficiency (Luo *et al*., 2011). The predomination of software reuse in the mainstream research of software engineering enables many large-scale and systematic software reuse, under the mechanisms of composition and abstract operation, to generate hierarchical architecture that is in accordance to the knowledge hierarchy of the application domain.

As a reusable unit in all the stages of software development process, the component includes system

Corresponding Author: Li-Guo Duan, College of Computer Science and Technology, Taiyuan, China

analysis, document, class, package and so on. Component model is an abstract description of the essential characteristics of the component, always regarded as one of the key factors influencing the successful realization of software reuse.

Although, a lot of research results have been obtained since the proposition of the concept software reuse, yet there has been a dearth of successful software reuse project in practical software production, due to the constraints of the analysis and design of reusable components. Software reengineering is such a process that changes the old legacy system to a new one with better performance that fit custom requirements and is easy to manage and maintain (Jie, 2011). Aimed at reuse issues in the legacy system of social security service platform, this study proposes a hierarchical model which is based on component reuse and consists of presentation layer, business logic layer and persistence layer. It also investigates the methods and procedures of software reengineering from the perspective of software reuse and puts forward a reuse model based on the hierarchical architecture.

RESEARCH APPROACH

Aimed at the legacy system of social security service platform, this study investigates the software reengineering of legacy system and proposes a reuse model based on the hierarchical architecture. The research approach of this study can be illustrated as follows in Fig. 1.

Method and technical route of legacy system reengineering:

- Research into the extraction of the logic in key businesses and reusable components

Software reuse extracts common features across software, for example, the method of software product line focuses on domain specified reuse. Data mining and reduction will result in useful components. The extraction of logic in key businesses and reusable components is for software reuse. Since reusable components must possess certain domain oriented generality, the first type of problems that needs to be tackled includes how to extract components and how to define the function of component. The issue of component extraction from legacy system can be studied from the following aspects: to extract object information from process oriented legacy code, to extract component information based on object information, to pack process oriented legacy system into component, as well as component metrics.

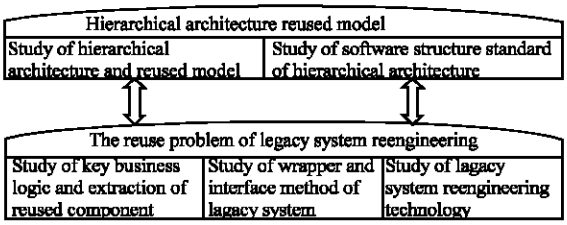

Fig. 1: The research scheme of lagacy system reengineering based on hierarchical architecture reused model

Drawing on the work of domain engineering, application engineering and component-based development, as well as the advantages of evolution-model and object-oriented development approaches, this study proposes an information system development model, which is based on software reuse and consists of three layers, i.e., demand reuse, design reuse and component reuse. The realization process is presented in Fig. 2.

- Research into the methods of legacy system packing and interface

With the gradual mature of object-oriented analysis and design, as well as the frame and component technology in the middle of 1990s, packaging technology was introduced to the legacy system. Legacy system packing technology varies with different application platforms. For example, Java platform uses JNI (Java Native Interface), C/C++ platform uses tool-SWIG (Simplified Wrapper and Interface Generator) and makefile is used for compiler project management (Min, 2010). Among all the integration technologies of legacy system, object packaging technology is one of the most important methods. It integrates the legacy system into the new system by regarding the former as an interface of the latter, which will facilitate the visit of other components. All the demands details and technologies of how to realize these demands that are required by system response are contained in the packaging interface. Aimed at features of the legacy system project, JNI technology is adopted to realize the object oriented packing of function component.

- Research into the technology of legacy system reengineering

There exist many problems in the legacy system, such as out-of-date technologies, missing documents, disordered structures and so on. In order to improve the reuse degree of legacy system, the research of legacy system reengineering becomes an important subject

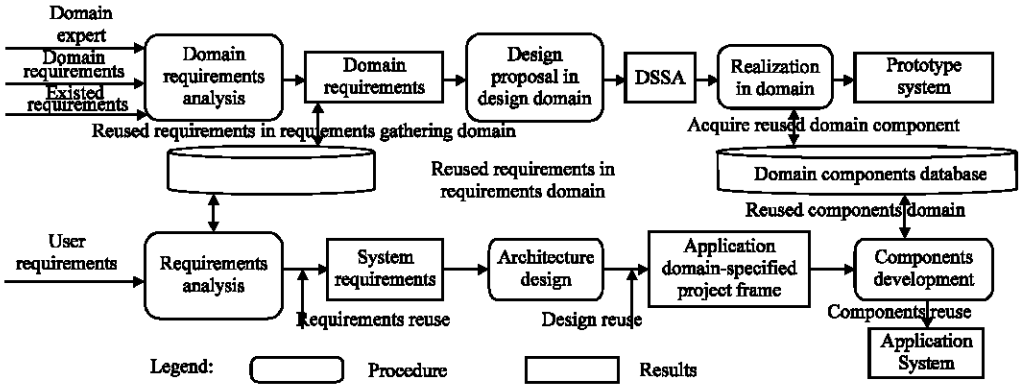

Fig. 2: Domain development model based on software reuse

(Xinyu, 2007; Li, 2008), whose implementation is facing many challenges. Firstly, great difficulty exists in the process of extracting component information from the legacy system and deploying these components in distributed environment. Secondly, it is difficult to present the extracted business logic at multi-layers and from various perspectives.

The reuse model method and technology route of the hierarchical architecture:

- Research into the hierarchical architecture and reuse model

At the same time when a layer in the hierarchical architecture provides service for its upper layer, it will also, as the client of the lower layer, call performance functions offered by lower layer. Besides, in certain cases, internal layers might open some services for the calls of other layers out of the treatment demands. In this way, each component in the hierarchical architecture forms virtual machines with different functional levels at different layers. Communication can be realized between virtual machines according to protocols; while the communication between non-adjacent layers is limited by some constraints.

Hierarchical architecture has many advantages. Firstly, it supports step-by-step abstraction in the process of system design; secondly, it has good expansibility and thirdly, it supports software reuse.

During the research of the reuse model of hierarchical architecture, attention should be paid to the issue of putting certain functions at certain layers, which is also one of the difficulties of hierarchical architecture.

- Research into the standard of software structure in hierarchical architecture

In the hierarchical architecture, the interface should be independent of the hardware platform that is used for service realization, operation system and programming language. Studies on the standard of software structure in hierarchical architecture mainly include features of the interface, availability of the operation, parameters, data type and access protocols. Another important research subject concerns the heterogeneity problems resulted from differences in programming language, in platform, in communication protocol and in data.

Implementation method and technology route of this study:

- To analyze the logic of key businesses and extraction of reusable components of the legacy system in social security service platform and to investigate the method and procedure of legacy system software reuse

Due to the large scale and miscellaneous businesses of legacy system, how to extract and describe key-business logic and reusable components become problems crying for solutions. The extraction of key-business logic and reusable components is for the realization of software reuse and its value will increase with the accuracy of the extraction of common properties and the height of extraction level. Therefore, aimed at characteristics of different types of legacy system, this study extracts key-business logic and reusable components through appropriate reverse engineering technology and method and investigates the method and procedure of legacy system software reuse.

- Aimed at the legacy system of current social security service platform, this study investigates necessary properties during the process of packaging, transplanting or redeveloping the legacy system, so as to implement its reengineering

Legacy system reengineering belongs to software maintenance. Aimed at the characteristics of different types of legacy system, this study combines forward engineering technology, remodeling and reverse engineering technology to package, redevelop or transplant the legacy system. It also investigates the method and technical features of legacy system reengineering so as to realize its reengineering.

- Aimed at the characteristics of legacy system and architecture knowledge necessary for software reuse, this study proposes a reuse mode based upon the hierarchical architecture

Belonging to design reuse, the reuse of architecture is a high-level abstraction of the system, reflecting major constituents and their interactions within the system. Therefore, it is more suitable for reuse. Since the social security service platform possesses above-mentioned characteristics and the reuse of the software architecture is practical, the proposition of the reuse model based on the hierarchical architecture will help with the construction of the reference model of social security service platform.

EXPERIMENT

Hierarchy can address the instable demands change of complicated system to make the design more expansible, flexible and logical. This study proposes a hierarchical architecture for the endowment insurance management system of the social security service platform. This architecture can guarantee clear segregation of functions and responsibilities, as well as a specified way by which components provide interfaces for external service and expansion. The application of the hierarchical model to the development of business component of the endowment insurance management system that is based on social security service platform improves the modular component performance and makes the whole system more effective, convenient and safe.

Component hierarchical model: In the hierarchical architecture, the further separation of client from the database of server makes the influence of changes in business logic on the client as small as possible. In short, the client, only responsible for collecting users' demands and displaying the treated results, does not care the structure of database any more. Aimed at the development of endowment insurance management system based on the social security service platform, this study proposes a hierarchical architecture that is based upon component reuse and consisted of three layers, namely presentation layer, business logic layer and persistence layer, as shown in Fig. 3. Being able to tackle problems related to the internal structure and organization of the component, this architecture guarantees clear segregation of functions and responsibilities, specified ways by which components provide interfaces for external service and expansion and also satisfactory expansibility and flexibility of the component.

Business logic design model under hierarchy model: The design of business logic model under hierarchy model is illustrated as Fig. 4. There are two parts (standard application and platform components) in this layer. The standard application part is realized by social security MIS kernel platform and it includes request process layer

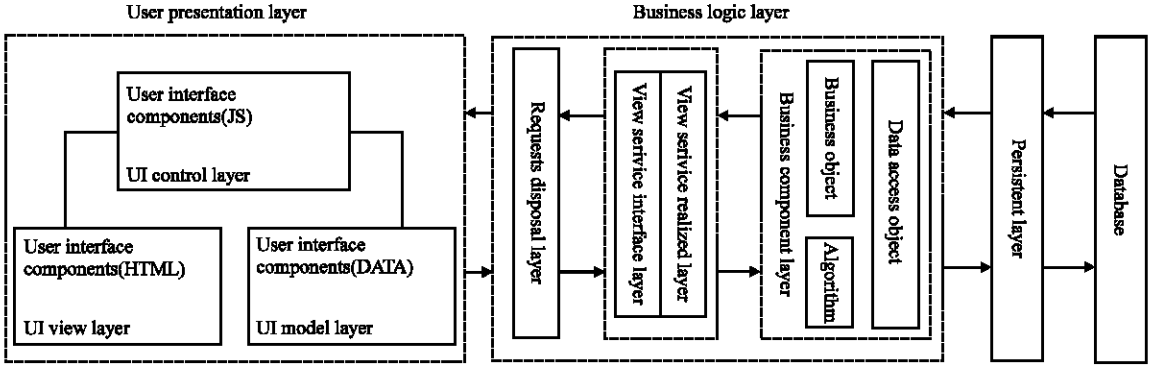

Fig. 3: Hierarchical architecture development model of social security information service platform MIS

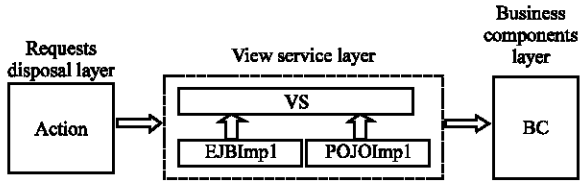

Fig. 4: Design model of business logic layer

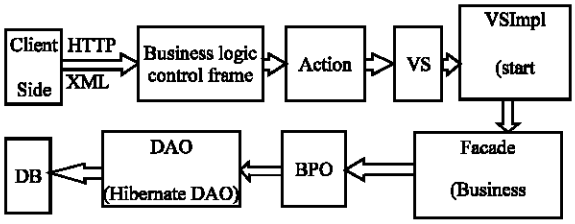

Fig. 5: Flow diagram under business logic

and view service layer. The request process and view service in this layer often need corresponding change and its reuse capability is lower when local business changes. The reuse parts in social security MIS kernel platform is mainly in business component layer. The business component can be invoked and combined by different view service layer to complete different business requirement.

Request process layer (Action) can invoke corresponding method in view service layer to complete the task. One business unit is correspondence with one interface in view service layer. The view service is divided into two parts, interface and realization. The detail realization technology can be hided by view service interface layer. The realization of view service interface can be POJO, or can be EJB and can invoke the detail business component in business component layer to complete business logic process. The system connects the system frame with business component layer through BusinessDelegate design pattern. The BusinessDelegate design pattern is just the Client of business component layer (POJO or EJB) and is the agent of component layer.

- Business component is the main reuse unit of platform. It includes a main business concept and provides a set of business services corresponding to main business concept. Business component can not only invoke method in persistence Layer to operate database, but also use operate class in persistence Layer in user platform. If there are requirements for HQL or SQL statements to complete some functions, corresponding DAO objects are increased to encapsulate the functions. If some computing strategies are needed in business component, the computing rules are needed to wrap in a single algorithm class

Component Software Reuse Based on business logic model: Software Component is the combining unit of software system and it shows the dependence to context by contractual interface. Component can be deployed independently in order to synthesize with the third party. Compared with traditional software entity such as modular and object, component has higher independence and integrity and self-description. From the point of view of application components, the project takes business components as the main reuse unit of platform and extracts several layers to provide convenient reuse development, such as business component Façade layer, algorithm object layer, business object layer, data access layer etc. The basic flow diagram is as Fig. 5 illustration.

- Business component Façade is the exterior interface of this layer and it is the exterior agent for all business methods within a business component package
- Business object is used for accomplishment of actual completed business logic. One business object can invoke other objects to process business and it can access database not only through persistent layer API, but also through invoking DAO object. The responsibility of PO object in business object layer includes: business logic realization; processing business logic centralized; complete the transaction processing
- The algorithm object is the encapsulation to different algorithm
- Data Access Object (DAO) is the encapsulation to database access. DAO layer is mainly made up by DAO object, business DAO objects, IPersistence interface, Platform factory class etc. DAO object includes data access logic rather than business logic and is responsible for interactive with database roundly by hiding realization details. Business DAO object accesses database through IPersistence interface and Platform factory class provides the factory method getPersistence to access IPersistence interface object. The realization of IPresistence interface encapsulates API in persistence Layer and provides the method to access database

CONCLUSION

In this study, we discussed the components reuse technology about legacy system reengineering and proposed a kind of hierarchical architecture reused model based on components aimed to the question of current social security information service platform. The model ensures components' extensions by providing standard and unified service interface and extended interface in normalizing and restraining the component internal structure. We applied the hierarchical component model in the old-age insurance MIS under social security information service platform and obtained improved modular component performance and more effective, convenient and safe system development. At present, software engineering based on component applied more in service oriented system development and in the future, the solution can provide more beneficial reference and help for legacy system reengineering during software development and integration in cloud computing platform.

ACKNOWLEDGMENT

This study was financially supported by the Science Research Project in Shanxi Province, China (20120321030), and Youth Foundation of Taiyuan University of Technology (No. 2012L067).

REFERENCES

Frakes, W.B. and K. Kang, 2005. Software Reuse research: Status and future. IEEE Trans. Software Eng., 31: 529-536.

Hui, Z., B. Kerong and W. Hongbo, 2010. Modeling and implementing of service-oriented reengineering process of legacy system. J. Naval Univ. Eng., 22: 25-30.

Jie, Y., 2011. Research and application on software re-engineering theory. M.Sc. Thesis, Anhui University, Anhui.

Li, W., 2008. Rapid conversion of the huge software legacy system. M.Sc. Thesis, Zhejiang University, Zhejiang.

Liu, Y., 2012. The study on software reuse based on service composition and dependency-aware service oriented architecture. JDCTA, 6: 161-168.

Luo, S., P. Zhili and Z. Changhai, 2011. A feature of reuse method at requirement level based on aspect encapsulation. J. Comput. Res. Dev., 48: 1714-1721.

Min, W., 2010. Research and implementation of a tool for system migration. M.Sc. Thesis, Xidian University of Technology, Xian.

Ximing, Z., 2010. Golden insurance project construction and future conception. E-Government, 2-3: 131-137.

Xinyu, W., 2007. Research on software reengineering of large-scale legacy system. M.Sc. Thesis, Zhejiang University, Zhejiang.