

Implementation of After-sales Service System based on SSI Improved Framework

Yongchang Ren^{1,a}, Yongzhe Ma^{1,b}, Hang Wang^{2,c}

¹School of Information Science and Technology, Bohai University, Jinzhou, 121013, China

²College of Engineering, Bohai University, Jinzhou, 121013, China

^a1213552916@qq.com, ^b1125444268@qq.com, ^cbhdxwanghang@163.com

Keywords: after-sale service; SSI framework improvement; system architecture workflow; function implementation

Abstract. In the highly competitive market environment, after-sales service has become the key for enterprises' survival. In order to solve the problem of difficult management of after-sales service, this paper surrounds framework improvements, system architecture workflows and business functions to construct the after-sales service system which is based on SSI framework. First of all, framework improvement processes are detailed introduced, improved methods on each layer are detailed elaborated; then, based on the improved framework, the system architecture workflow is implemented; finally, main business functions implemented by the system are described. The implementation of the system can centrally manage the repair, maintenance, installation and other daily work; standard after-sales service business processes, information manage spare parts, reduce the manpower, material resources and costs, which greatly improves the quality and management efficiency of after-sales service.

Introduction

After sales service refers to the related series services of delivery, installation and maintenance which are provided by manufacturers follow up after customers buy products [1]. Along with the continuous development of science and technology, and the globalization of market economy, the difference between products product by major manufacturers in performance, technology and other aspects is more and more small, only pay attention to product production and sales links is not enough to face the increasingly intense market competition, major enterprises gradually realize the importance of customer service [2]. Customer service is the key to customer relationship management [3], after-sales service plays a very important role in it. In daily after-sales service work, manually fill in the documents as the main working way, inevitably due to human factors, the problems of information transmission error and spare parts application delay occur, which lead to maintenance personnel can't arrive at the spot in time to maintain equipments that delays responses time, reduces working efficiency, brand influence and customer satisfaction are directly affected. In addition, because the data cannot be stored for a long time, when install equipments, installation personnel cannot quickly and comprehensively grasp the related detailed information of the equipment, and do not know apply which kind of spare parts for installing, which greatly reduces the installation efficiency. Therefore, it has become inevitable that enterprises speed up to construct a complete set of after-sales service management system.

In view of the shortcomings of the original SSI architecture, this paper makes improvements on the basis of this, draw the interface layer in it, based on the design document, surrounding the SSI improvement framework, system architecture workflow and business functions, implementing the after-sales service management system to provide referenced information and ideas for enterprises build the same type of after-sales service systems in the future.

SSI Framework Improvement

Functions of struts, spring and ibatis framework are powerful, but its advantages are only limited in

one layer of the development system structure, which are unable to complete actual development tasks. In view of existing problems of SSI architecture, this paper uses the MVC design pattern, draws the interface layer to management data manipulation interface, and combines with the characteristics of three kinds of frameworks to integrate reasonably to improve the existing architecture. Methods are as follows: with the struts framework as the infrastructure, fuse the spring framework to build in the presentation layer to receive requests send by users and the data input by users to complete the system interact with users; The integration of the spring framework and ibatis framework is built on the business logic layer, the data operation CRUD interface provided by the interface layer to complete handles of business logic and transaction; the ibatis framework is built on the data persistence layer to complete the system interact with the database. A seamless combination of three frameworks, full combination with their respective advantages. SSI fusion architecture is shown in Fig. 1.

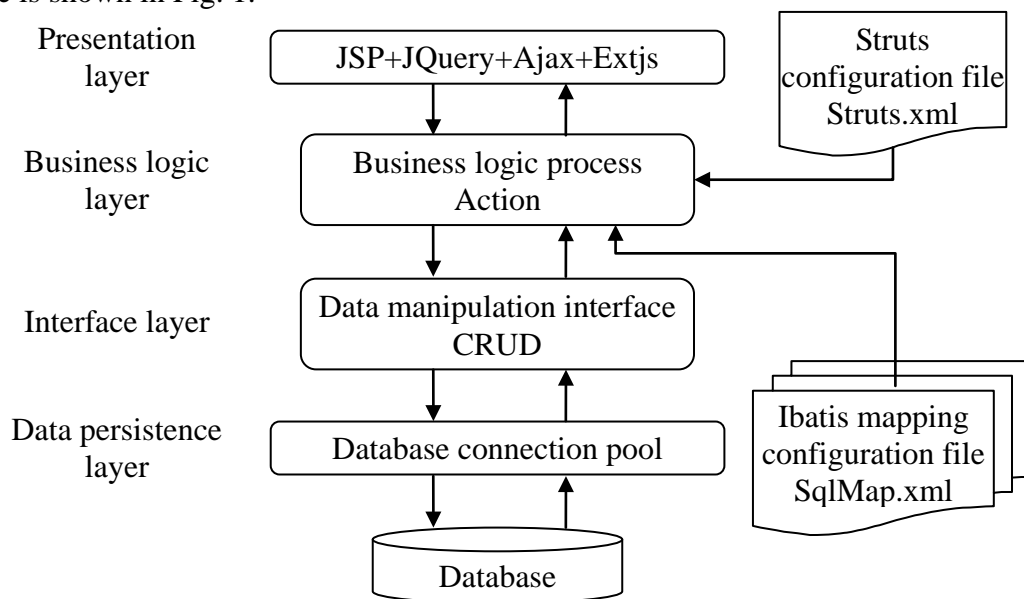


Fig. 1. Fusion structure diagram

Presentation layer fusion principle. First, user requests, using Ajax technology to realize users asynchronous request Action class; second, load the Struts framework, "ActionServlet" is configured in web.xml to load resource configuration files; request intercepting filter configuration is configured to get the user request, address parameters ends with the forms of ".do, .action"; third, load resources configuration file, the Action class and related request pages are configured in it to save the corresponding relationship.

Business logic layer fusion principle. First, interface class injection, first of all, interface classes configuration, configuration rules are as follows: the attribute "Id" is the name that "Action" class retrievals interface; properties "class" is the implement class defined by interface, it must be declared the absolute path in the project; then, Spring reverse control is used to inject business processing objects; second, Struts integrates with service layer, indicate the interface name to obtain corresponding service components from the spring container.

Interface layer fusion principle. Define the system level data operation interface and implementation classes, and provide function interfaces such as add, modify, delete and query for the business logic layer.

Data persistence layer fusion principle. First, custom data manipulation XML files, and mapping the corresponding service implementation class; second, load files in the Spring resources container.

The specific integration implementation of each layer is as follows:

Presentation layer. Struts framework uses the MVC mode, pay attention to isolating the view layer and control layer, but not involve the optimization design on the model layer [4]. After receiving users' requests of jsp pages, retrieves "class" attribute of corresponding action tags in the struts.xml configuration file, combines with inversion control mechanism of the spring framework

to get the business logic processing interface corresponded to attribute values, specific business logics are handled by the business logic layer, struts is responsible for returning processing results back to corresponding jsp pages to display to users [5]. Developers need to configure struts.xml, according to business write logic processing classes. Adopting this method can optimize the model layer effectively and reduce the coupling degree between layers.

Business logic layer. Replace the remote call technology in the original SSI architecture, adopting the "interface and implementation" method, depending on business logic implementation classes provided by spring's control inversion container, combining with aspect oriented programming (AOP) technology deals with specific transactions and realizes the separation of business logic; its object mapping relation technology fully supports the ibatis framework, keeps loose coupled associations between entity classes with data table structures, through the transfer of data operation interfaces of the interface layer, implements the data interaction and returns results back to the presentation layer [6]. Developers need to write and inject business logic interfaces and implementation classes according to actual needs.

Interface layer. Provide the unified data operation interfaces of the transaction processing implementation classes for the business logic layer, including retrieve, create, update and delete, personalized data operation functions can be finished according to any business scenarios. Developers need to write and inject interfaces and implementation classes.

Data persistence layer. It is the implementation of the "Dao" interface in the interface layer, which is responsible for communications between a database management system to provide data access services for the business logic layer [7]. Ibatis has a semi automated object relational mapping function, adopting the SQL free writing way in xml configuration files implements data operations of Dao interface objects, which completes the persistence of data interaction [8]. During this period, both are independent of the mapping layer, and not affected by each other. Developers need to configure ibatis SQL mapping files, and inject them into their management files.

System Architecture Workflow

When users login system, authentication is required. Only the user name and password input are completely accurate, users will enter the system; otherwise, the system reminds that input user name or password is error. This paper uses accounts authentication as an example to describe the system architecture workflow, the specific data flow is shown in Fig. 2.

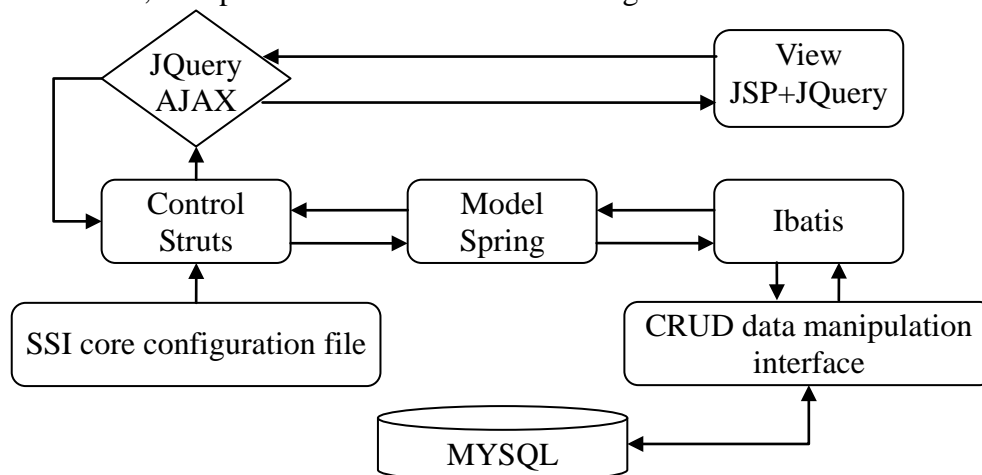


Fig. 2. User authentication function implementation flow diagram

Description: SSI core configuration file is used to configure web.xml and struts.xml files to locate the request path "url= login.do?reqCode=login" to LoginAction of the control layer;

Step one, JQuery check whether the user name and password are empty; if empty, the system prompts users to input; if not, the system retrieves the user login information;

Step two, sent request path "url=login.do?reqCode=login" through post functions, the request contains user name and password;

Step three, LoginAction, organizationService and organizationServiceImpl are injected into Spring container, organizationService is injected into LoginAction to call getUserInfo interface;

Step four, getUserInfo interface is realized in organizationServiceImpl;

Step five, Ibatis returns processing results to the business logic layer;

Step six, Spring receives processing results to return to LoginAction;

Step seven, according to processing results, struts forms information;

Step eight, through post callback function, processing results will be displayed to users.

As can be seen from the above figure, this paper improves the SSI framework, which is based on the existing three layer architecture to be extended to four layers, the presentation layer, business logic layer, interface layer and data persistence layer. The presentation layer obtains the account information input by users and send a url request to the business logic layer, business logic layer receives the request to query user information through calling the "getUserInfo" method of the interface layer, and returns results to "organizationServiceImpl", Spring container captures data to package to "LoginAction" to verify user account, and returns data validation results back to the presentation layer in the form of "Json", Struts skips to the corresponding view for users to operate according to the return scene.

Function implementation

Each module function implementations of the system are carried out around spare parts types, business operations and procedures of different spare parts have large differences. Based on the requirement analysis and system design result, it is necessary to implement six management modules' functions, involving basic data, spare parts application, spare parts management, equipment maintenance and inventory, main modules' functions implementation process are described as follow.

Basic data management. Record five basic information, implementation operations have add, update, submit and batch delete. Among them, contract information and equipment models can be freely operated, not need to submit for approval; after spare parts and equipment information are submitted, the data are effective and can't be deleted and updated, among them, installation equipments are submitted to secretaries to approve, spare parts information are submitted to sales supervisors to approve, After consent, information become effective; otherwise, documents are returned to be updated to submit again.

Equipment maintenance management. Maintenance personnel receive repair orders to maintain equipments, therefore, the module is composed of two parts, repair and maintenance work order management. Repair work order management mainly carries monitoring on the equipment maintenance schedule, at the end of the maintenance, repair work orders unilaterally end; maintenance work orders save detailed information of maintenance equipments, which indicate spare parts consumption and return details. Two parts' concrete implementation process and key points are as follows:

First part, repair work orders management. landing system to complete repair work orders with the customer service role, support basic data manipulation. After adding work orders, document status is draft, it is time for batch delete and update; when submit, choose sales supervisor for approval, document status is submitted at this time, only views the data; documents return to sales supervisor, who can update documents contents, also reject directly, customer service personnel can resubmit after updated, after consent, dispatch maintenance personnel; there are three choices after maintenance personnel receive documents: accept, work orders are complete, repair work orders are over; don't accept, sales supervisor need to assign other maintenance personnel; if there are emergencies, maintenance personnel can also receive repair orders to sent to other personnel.

Second part, maintenance work orders management. Login system to complete maintenance work orders with the maintenance personnel role, support add, update, review, delete, submit and export functions. After adding work orders, document status is draft, it is time for batch delete and update; when submit, check whether the sum of spare parts consumption and return quantity is equal to the number of applications; inconformity, prompt spare parts return applications,

documents are not submitted. After submitted, document status is submitted, it's time for only viewing the data, documents are returned to sales supervisor, if agreed, document status is completed, operations are over; if not, documents are returned to maintenance personnel, document status is back, not deleted, after work orders contents are updated, they are submitted again.

Conclusion

This paper draws the interface layer into SSI framework to realize the improvement, the coupling of the code is reduced, the system's reusability and robustness are enhanced, which accelerates the system's development, reduces the late maintenance capacity, effectively solves the problems which exist in the original framework [9], comprehensively enhance the quality of the system. Around the system architecture workflow and business functions, complete the construction of after-sales service management system which centrally manages the repair, maintenance and other daily work; standardizes after-sales service business process, and realizes spare parts information management, reduces costs, manpower and material resources; which greatly improves the quality and management efficiency of after-sales service, can be more convenient to service for customers, deepen customer's impression of the product, and improve customer satisfaction with the product. It is of great significance for promoting the brand influence and enhancing enterprises' own market core competitiveness [10].

References

- [1] S. Murali, S. Pugazhendhi, C. Muralidharan. Modeling and Investigating the Relationship of After Sales Service Quality with Customer Satisfaction, Retention and Loyalty—A Case Study of Home Appliances Business[J]. *Journal of Retailing and Consumer Services*, vol. 30, no. 1, pp. 67-83, 2016.
- [2] Lucie Kanovska, Eva Tomaskova. Trends in Customer Services and Inter-functional Coordination by Manufacturers[J]. *Procedia-Social and Behavioral Sciences*, vol. 213, no. 1, pp. 677-682, 2015.
- [3] Sartika Kurniali, Titan. Customer Service Information System for a Call Center[J]. *Procedia Computer Science*, vol. 59, no. 1, pp. 298-304, 2015.
- [4] Z. Y. Dong, "The Sheme of Adding Slot Ports to Resource Management System Based on Struts2 Ajax Validation," *Journal of Zhengzhou University of Light Industry: Natural Science*, vol. 28, no. 4, pp. 92-96, 2013.
- [5] W. L. Gan, W. C. Liu, "Development of Enterprise Web Application Based on Struts2 and Ajax," *Industry and Mine Automation*, vol. 39, no. 2, pp. 23-26, 2013.
- [6] X. L. Zhao, "The Application of Spring Timer," *Journal of Taiyuan Normal University: Natural Science Edition*, vol. 14, no. 4, pp. 48-53, 2015.
- [7] J. Meng, "Proficient Spring-Java Lightweight Architecture Development Practice," Beijing: People's Posts and Telecommunications Press, vol. 10, pp. 278-279, 2006.
- [8] M. N. Liu, "Research on Application of SSI Framework in Teaching Resource Platform," *Microcomputer Applications*, vol. 32, no. 2, pp. 43-44, 2016.
- [9] C. M. Chen, "Analyze and Review on The Measurement in Object-oriented Software," *Journal of Changchun University of Science and Technology: Natural Science Edition*, vol. 36, no. 3-4, pp. 152-156, 2013.
- [10] Chi-ming Ma, Tai-jung Wu, Shio-Yang Wu. A Rule-Based Approach for DynamicAdaptation of Service Flow. In: 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA). Taipei, 2012, 1-4.