

# Research and Implementation of Construction Company Management Information System Based on Microservice and Service-Oriented Architecture

Jing Gong, Caifeng Cao\* and Tongtong Su

Computer School Wuyi University Jiangmen, Guangdong, China

\*Corresponding author

**Abstract**—This paper explores the application of micro service and service oriented architecture in management information system of construction field. The use of micro services to implement functional components and through the integration of different platforms and heterogeneous micro services, the construction company management system is realized, which greatly improves the efficiency of enterprise management, enhances the core competitiveness of enterprises, reduces the cost of system reconstruction, system maintenance and upgrading.

**Keywords**—software architecture; service oriented architecture; microservice; web services

## I. INTRODUCTION

With the continuous development and expansion of enterprises, innovation of management and business pattern has become an important factor in enhancing the core competitiveness of enterprises. However, the current service-oriented applications of construction enterprises are mostly monolithic applications. The core of the applications is ESB (enterprise service bus). Generally, ESB has a very complex structure, which is not conducive to the expansion of new features of the application and may cause complicated system maintenance [1]. Research and practice of micro-service and building micro-service and service-oriented architecture based application system are of great significance and guiding value for agile development, function expansion and reconstruction of construction enterprise application system.

## II. SYSTEM PLATFORM ARCHITECTURE DESIGN

System architecture is based on SOA (service oriented architecture), in order to improve the reusability and expansibility of the software. Adopting the method of "horizontal and vertical segmentation", this paper designs the system in a hierarchical way, modularize the functionality of the system and encapsulate business functions into services. The system architecture is divided into six layers, which include UI interface layer, business logic layer, service layer, data access layer and data storage. The overall framework of the system is shown in Figure 1.

UI layer is also the presentation layer, which is an intuitive interface between user and system. It is responsible for obtaining the user input parameters and receiving the user's various event processing requests.

Business layer is designed according to the functional requirements of the system and is responsible for various business logic processing. It includes modules like customer management, contract management, project case management, news management and system management.

Service layer encapsulates the common business logic into a service and provides interfaces for the system integration. The service in this information system is divided into two categories, one is customized functional service, and the other is third party service. Self-defined services include customer management service, employee birthday reminding service, information updating service. The third party service called in this system is weather forecasting service which is provided by www.WebXml.com.cn.

Data access layer controls all kinds of access operations related to the system data, and provides data access interface for the business logic process, and thus to achieve the purpose of reducing the frequency of database interactions and accelerating the speed of application execution.

Data Storage layer is mainly used to store system and service data. The system uses ado.net, jdbc components and mybatis to cooperate with each other to ensure the security and consistency of data. According to the system platform, SQL Server 2008 is selected for the system, information updating service data storage using SQLite database. Services use json format data in the data transmission process.

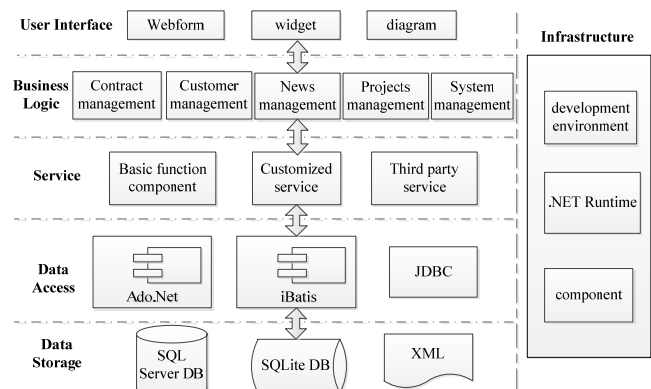


FIGURE I. THE OVERALL ARCHITECTURE OF THE SYSTEM.

### III. SYSTEM SERVICE DESIGN

The service of the system is constructed by micro service. There are great differences in the development, deployment, monitoring and maintenance between micro services and services in traditional SOA architecture. The main features of the micro service as follows:

- Using services to achieve componentization. Business functions are implemented by micro services instead of components. Micro services are called relatively simple and convenient.
- Service independently manages related business and data maintenance. Each service can have its own data storage, suitable database is selected according to the specific business needs and provides interfaces for data calls.
- Each micro service encapsulates its own business logic independently, and service can be hosted in different host environments.
- The implementation of micro services includes the front-end and back-end as well as the deployment and maintenance, covering almost the entire life cycle of the service.
- The service is fine-grained. Service granularity represents the functional scope of the service. Business processing logic of fine-grained service is simple, which refers to the SRP (single responsibility principle) principle. The application is decomposed into a group of micro services and will not change the overall function at the same time.

In the development process of this system, division of micro service is based on specific business requirements and services can be independently deployed on a server or other host environment, which can achieve the independence of service development, testing and deployment [3]. Through the isolation of the service process, the system failure will be limited to local or single service, will not affect other functions of the system, so as to ensure that the system's overall available, local fault tolerance. In addition, the development language, platform environment and data storage of services can be selected according to the actual situation. In this system, WCF is adopted to build micro service including customer management service, employee birthday reminding service and information updating service. The modeling design is as follows:

#### A. Customer Management Service

It is based on .NET platform. The service contract class is Client Management Service. cs. Service contract is identified by the attribute [Service Contract]. Client management service defines the interface methods include Get Client List(string str), Get Feed Back Info By Id(string fid), Del Client Info (string str), update Client Info(string str), add New Feed Back(string str), add New Client (string str), etc. These operations of the service are identified by the attribute [Operation Contract].

#### B. Employee Birthday Reminding Service

The employee birthday reminding service defines interface methods include Get Result (string birthdate), which is identified by the attribute [Service Contract]. The service interface contract class is IBirthday Reminder Service. cs.

#### C. Information Updating Service

It is based on the Java EE, the JAX-WS (Java Xml Web Services) is used to implement the service that is based on XML SOAP messages [2]. The service defines the interface methods include dele Info(int id), get Info List(). The service interface class is I Update Info Service. cs. The service database is built with lightweight database SQLite to store information obtained by the open source crawler tools.

### IV. SERVICE IMPLEMENTATION AND DEPLOYMENT TESTING

#### A. Service Implementation And Deployment

The customer management service and employee birthday reminding service implemented in the system are hosted on the IIS server. The service configuration file is web. config, which contains the node <system. Service Model>. That node specified service endpoint, which contains transport protocol binding, service contract, connection timeout, configuration file, hosting environment settings and other information. The endpoint of the service contains the information needed to implement service communication, which can be accessed by service endpoint address. A service can be configured with multiple endpoints. Service binding uses HTTP-based ws Http Binding which supports interoperability and enables that the service client can work with the server. The security of transport layer uses IIS security mechanisms. The details of each service are shown in table 1.

Functions of customer management service include adding and deleting customer information or customer feedback, modifying customer information, classifying the customer by area, consumption level or industry, as well as checking customer feedback and the status of feedback processing. The service implementation class is Client Management Service. svc.

TABLE I. SERVICE DEPLOYMENT DETAILS

Service Name	Details			
	Language	Binding Protocol	Server & port	Endpoint url
ClientManagementService	C#	Ws HTTP Binding	IIS7.5/8095	http://10.10.101.242:8085/BirthdayReminderService.svc
BirthdayReminderService	C#	Ws HTTP Binding	IIS7.5/8085	http://10.10.101.242:8085/BirthdayReminderService.svc
UpdateInfoService	Java	Basic HTTP Binding	Tomcat 7.0/8080	http://10.10.101.242:8080/InformationService/UpdateInfoServicePort

Employee birthday reminding service can judge whether it is birthday and return the results based on user's birthdate. The service implementation class is Birthday Reminder Service. svc.

Information updating service is written in Java, which JAX-WS2.0.jar and jdbc.jar is needed. Its functions include getting information list and deleting information. The service implementation class is UpdateInfoService.java. Service endpoint is Update Info Service Port, service proxy class is UpdateInfoServiceDelegate.java. Configuration files include web.xml and sun-jaxws.xml, which contains WS Servlet instance, WS Servlet Context Listener instance, endpoint and service implementation class, etc.

The running state of system called services is shown in Figure 2.

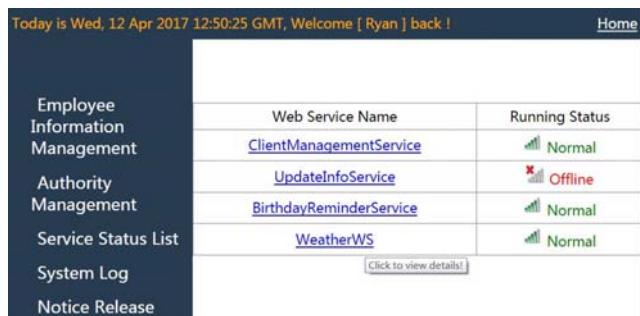


FIGURE II. SERVICE STATUS LIST

### B. Service Testing

The goal of service testing is to ensure service availability, integrity, accessibility, security, and reliability. The open source, cross platform Web services testing tool Soap UI is used for services testing in the system. Soap UI is a GUI automated development tool developed by standard Java Swing, which supports multiple platforms. Its functions include service function call test, security verification, regression test, load performance test and compatibility test. It allows users to complete service interface test through the WSDL of Web services, such as SOAP-based Web services, or REST-style Web services.

After installation of Soap UI 5.3.0, a new test project is created. In this test, the information updating service is chosen as the test case. SOAP message request based on HTTP post is shown in Figure 3.



FIGURE III. SOAP REQUEST MESSAGE

After the service receives the request, the response message returns with the correct json string information data. The detailed test result is shown in Figure 4.

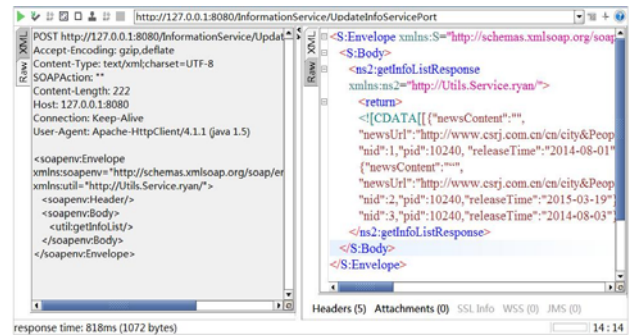


FIGURE IV. SERVICE RETURN VALUE

And then soap UI is used for load performance testing of the test case. The test scenario is 100 concurrent users, test time is 3 minutes, and test delay is 0-0.5 seconds. Results show that the average response time is 62.44 milliseconds, the total number of transactions is 283387 times, and the error rate is zero. Load test results are shown in Figure 5.

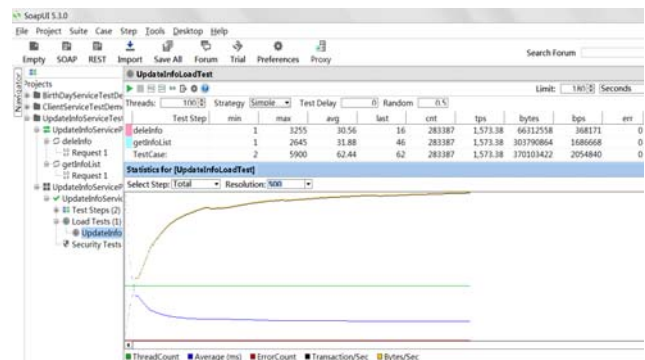


FIGURE V. LOAD PERFORMANCE TEST

Other services are tested separately by the same way, service deployed on the server returns data accurately and service performance is reliable, which ensures the normal operation of the system.

## V. SUMMARY

This paper puts forward the solution of construction enterprise management system based on micro service and service oriented architecture. Through the micro services to achieve the modularization, business functions with high reusability are encapsulated into services that can be deployed independently and called cross-platform or language. As micro service can be deployed independently, the services run in different processes and are isolated from each other. Business function changes mean modifying the corresponding services, and then redeploy the services. And the deployment of services for the system will not affect other functional modules. This solution satisfies the demands of the construction enterprise information management system by building loosely coupled, composable and scalable system architecture.

## ACKNOWLEDGMENT

I would like to express my gratitude to all those who have helped me during the writing of this thesis. I gratefully acknowledge the help of my supervisor professor Caifeng Cao. I do appreciate her professional Suggestions during my thesis writing. Also, I would like to thank Miss Na YANG, who kindly gave me a hand when I was doing the theoretical research and system development. At last, my gratitude also extends to my family who have been assisting, supporting and caring for me all of my life.

## REFERENCES

- [1] Mi Deji. (2009). Research and application of SOA architecture based on ESB.(Doctoral dissertation, Nanjing University of Aeronautics and Astronautics).
- [2] Wang G. Improving Data Transmission in Web Applications via the Translation between XML and JSON. Third International Conference on Communications and Mobile Computing. IEEE, 2011, pp. 182-185.
- [3] Martin Fowler, James Lewis. "Microservices", unpublished.
- [4] Jae Choi, Derek L, Nazareth. Implementing Service-Oriented Architecture in Organizations. Journal of Management Information Systems, 2009, vol. 26, pp. 253-286.
- [5] Guo D,Wang Wei, ZENG G S. A New Cloudware PaaS Platform Based on Microservices architecture. Netinfo Security, Nov 2015, vol. 11, pp. 15-20.
- [6] Vianden, Lichter and Steffens. Experience on a Microservice-Based Reference Architecture for Measurement Systems. Asia-Pacific Software Engineering Conference, 2014, pp. 183-190.
- [7] Rahman M, Gao J. A Reusable Automated Acceptance Testing Architecture for Microservices in Behavior-Driven Development. Service-Oriented System Engineering (SOSE). IEEE, 2015, pp. 321-325.