Research on Reuse of Legacy System Based on SOA

Jian-long Li¹,a, Pan-qing WANG² and Qi-yu HAN³

Department of Information Engineering, Mechanical Engineering College,
Shi Jiazhusng 050003, Hebei, China

a360689880@qq.com

Keywords: Legacy system, SOA, Web services.

Abstract. At present, with the country to promote the optimal combination of various resources, many systems are expanded and merged. New sections, new features and information systems are expanded. The new information system and the original information system cannot be effectively integrated. A large number of key information need to be shared in a relatively independent system. How to integrate the existing legacy system into a reusable Web service component. Integrating the isolated information system, eliminating the "information island" phenomenon, and realizing the reuse design of the legacy system. It is the construction of all kinds of information integration platform of the urgent problems to be solved.

Introduction

With the development of the time and technology, some of the legacy systems used for many years which in their interface and business processes are gradually unable to meet customer needs, but many areas of the original system have a strong dependence. It is imminent to redefine and upgrade similar legacy systems through software reengineering. Legacy systems are both problematic and practical. Existing problems: running on slow and difficult to maintain the hardware, the lack of documents led to software maintenance costs are extremely high, the system lacks easy to operate and modern interface, not scalable, running in a stand-alone environment, lack of network access characteristics and network publishing interface, difficult to achieve resource sharing. Not in accordance with the idea of stratification or component design and development, so most of them are relatively isolated systems, the lack of integration with other systems of information, and interface code is usually associated with business logic code and data access logic code mixed together Write, code readability and maintainability are poor. The practical value of legacy systems includes a large number of effective business transaction models and business rules that are impeccable in logic and algorithms and are irreplaceable in business. The original concept of SOA was proposed by Gariner in 1996. Because the technical level and market environment at that time did not yet have the real conditions for implementing SOA, SOA did not attract people's attention and SOA remained silent for some time. Web services began to pop, the Internet quickly appeared on a large number of different platforms and language development based on the identification of Web services components. In order to be able to effectively manage these large numbers of components, there is an urgent need to find a new service-oriented distributed Web computing architecture. SOA architecture to enable these different organizations to develop Web services to learn from each other and interact, to ensure security and balance reuse and manageability. As a result, people re-find the SOA architecture, and give the characteristics of its times. This paper proposes a framework of system information
integration based on SOA, which solves the shortcomings of the integrated system of traditional information integration technology, such as easy maintenance, lack of scalability, dynamic adjustment and narrow application range.

SOA Framework
The Basic Concept
First, SOA consists of a series of business-related services which can be used to achieve business goals, which can be set in a combination of multiple applications and invoked, through Internet-based public standards and protocols. SOA consists of three basic components: service providers, service intermediaries, service consumers. The relationship between the three is shown in Figure 1:

![SOA Components Diagram]

Figure 1. SOA components and the operation between each other.

Service provider: The service provider creates a Web service and publishes the interface and access method of the service at the service broker. Service providers need to decide which services are public in the process of publishing services, how to balance security and ease of use, and how to price services. Service providers decide the classification of services. The protocols are required to use these services.

Service intermediaries: service intermediaries are known as the service registry, service intermediaries is responsible for all potential service consumers which can see these Web services interface and the use of information. Service intermediaries need to define their own services, the scope of services can be large enough on the Internet, it can only be limited to a number of users. In addition, the depth and breadth of the information provided by the intermediary also need to decide, some intermediaries may provide a particular industry services, and some will provide a high degree of credibility of the service. The intermediary is classified Intermediary catalog.

Service Consumers: The service consumer needs to locate the entry at the service broker through various search methods and to contact a service provider and invoke the Web service it provides. Service consumers face the question of how to choose to provide quality service providers and assess risk issues.

SOA Features
SOA is a coarse-grained, loosely coupled service architecture that communicates through a simple but precisely defined interface and does not involve the underlying implementation and communication model of the underlying programming. Compared
with the traditional system architecture, SOA architecture has the following characteristics:

1) The reuse of services. In order to achieve reusability, the service operates only in the context of a particular process and is independent of the underlying implementation and changes in customer requirements. The reusability of the service can significantly reduce the development costs, allowing the legacy system to re-play its original value without having to re-build from the beginning to achieve the purpose of effective protection of corporate IT assets.

2) Service package. SOA architecture, through the legacy system components can be packaged into a business function and be used for business processes, reusable services, in the new application system to call. It implements information or simplifies the change in business data from an effective, consistent state to another, while at the same time hiding complexity.

3) Interoperability of services. In SOA, services are interoperable through established communication protocols, where synchronization and asynchronous are two of the main communication mechanisms. The interoperability features of the services provided by SOA can be reused in more cases.

4) The size of the service. Service granularity can be divided into fine-grained services and coarse-grained services, which refer to the scope of services to the outside world. Fine-grained service is a service that provides a small number of business processes, while coarse-grained services are those that provide specific business functions and propose research and implementation of legacy systems based on the B/S architecture for high-level business logic service. Fine-grained services cannot solve specific business problems, but they are more cohesive and more reusable and versatile.

5) The loosely coupled service. The service is loosely coupled, and in simple terms, the binding and service between the service requester and the service provider, the two should be loosely coupled, that is, the service requester will not know what the service provider is Specific technical details, such as programming language, running platform and so on. Loosely coupled is the most important feature of the SOA architecture and can be used to differentiate it from other component architectures.

Legacy System Integrated Frame

The Basic Requirements of Legacy Systems Integration

Taking into account the characteristics of the legacy system itself, to achieve the integration of legacy systems information will be subject to many restrictions in the realization of technology should meet the following basic requirements:

1) The information integration of the legacy system should be the integration of the specific application of the legacy system, and should support the normal interaction of the information required for the normal operation of the business, rather than the simple interconnection and information exchange of the legacy system.

2) According to business strategy and demand changes in a timely manner to adjust the integration, Integration is dynamic;

3) Should fully consider the new business on the safe and stable operation of the requirements to ensure that the legacy subsystem in the integration can still maintain a separate operation of the security and stability;

4) The information integration of the legacy system should be an extension and extension of the functionality of the legacy subsystem, rather than overthrow all the functions of the legacy subsystem.
Legacy System Reuse Architecture

SOA and the traditional system construction method is different, it is not entirely application-centric, but to service-centric construction. The service is coarse-grained and loosely coupled. WSDL design concept completely inherited the XML-based contemporary Web technology standards consistent design concept: open. WSDL allows the use of other types of definition language, allowing the use of multiple network transport protocols and message formats. At the same time WSDL also applied the reuse of modern software engineering concept, the separation of the abstract definition layer and the specific deployment layer, making the abstract definition layer greatly increased reuse. For example, you can use the abstract definition layer for a class of Web Services abstract definition, and different operating companies can use different specific deployment layer description with abstract definition Complete its own description of Web Services. Therefore, WSDL to describe Web Services package service interface is the basis for enterprise legacy system reuse. Figure 2 is based on the SOA architecture of enterprise legacy system reuse framework.

(1) Legacy systems including existing CRM, SCM and ERP packaged applications, as well as some older object-based system implementations, business intelligence applications, and legacy system data.

(2) Service component layer, reuse is an important principle of SOA thinking. So, in the SOA architecture of the architecture must be fully reuse the existing application
system. In order to integrate these existing applications in a loosely coupled manner. You can encapsulate existing information systems into Web services so that they expose interfaces in a unified way WSDL. The interfaces that were exposed in various APIs were re-described with WSDL and then used http or soap messaging as a bridge to interact with the outside world. The interface encapsulated with Web Services can mask the implementation details of the original system and eliminate the difficulties of integration between different technologies. The Web Services encapsulation enables external applications to use system services in a uniform, loosely coupled manner. When the business logic of the system needs to be changed, as long as the WSDL describes the interface unchanged, regardless of the business logic of the system, the implementation of the technology and even the replacement of the new application system, the client does not need to make any changes. So the service component layer will be left behind the system in accordance with the appropriate granularity decomposition and then use Web Services to be packaged into service components.

(3) ESB is the development of past message middleware, which uses a bus-like model to manage and simplify the integration topology between applications, based on widely accepted open standards to support applications between messages, events And the level of service on the dynamic interoperability. The ESB abstracts the application components into a set of services that are reusable and interface-defined services that communicate through the bus (rather than direct point-to-point communication interactions). ESB focuses on two important issues in SOA, that is, the statement and discovery of services.

(4) The service found in the identified public service at this level. They can be found directly statically bound, then called, or organized into a composite service. The interface of this layer is exposed as a description in which they are made public for use. They can exist independently or as a synthetic service.

(5) Service composition and presentation in the service layer are defined in this layer. By matching, scheduling, services are bound to a process, usually using XML-based WS-BPEL language for business process description. The business flow of enterprises in SOA should be the assembly and sorting of coarse-grained services. The different combinations of services represent the different business processes of the enterprise, thus realizing the dynamic business model.

(6) An access or presentation layer which is the gateway to an enterprise business process performs a business process in the presentation layer by entering a message into the service flow engine in the business layer. Of course, in the process of reuse of the system also need to consider the quality of service, security and other issues, no longer elaborated here. In short, through the reuse of the framework can see the SOA architecture in the legacy system reuse process has a unique advantage.

Conclusion

In this paper, a brief analysis of legacy systems and SOA architectures is presented, and a reuse framework of SOA architecture is proposed. Based on this framework, some modules of legacy system are encapsulated by Web Services. Practice has proved that the SOA system based on Web services to achieve the legacy of a wide range of reuse, SOA architecture with its unique advantages in the enterprise information process plays an important role. Which solves the problem of enterprise resource sharing in the network environment, and solves the problem of reusing the legacy system on the Web?
References


