

A Legacy System Encapsulation Strategy Based on Web Service

D. J. Chen¹, X. Y. Li²

¹Dali Comprehensive Inspection Centre of Quality and Technical Supervision, Dali, P.R. China

²School of Mathematics and Computer Science, JiangHan University, Wuhan, P.R. China

Email: 652187048@QQ.com, Lixy_76@163.com

Received 2012

In order to migrate the enterprise legacy system to the web, a multi-agent based legacy system encapsulation model is proposed. Firstly, the characteristics of legacy system are analyzed, and then the data and functions that need to be published are confirmed. Secondly the legacy system is wrapped into web components with common interface, and these components are managed by the application server. Thirdly, the clients can send requests to the application server, and receive the return result from the application server. Due to adoption of wrapping technology for legacy system, original security and stability of legacy system are guaranteed in the web components. Finally, the validity and practicability of the migration technology are verified through the application in the encapsulation of Matlab as web components.

Keywords: Legacy System; Multi-agent; System Migration; System Encapsulation

Introduction

At present, there isn't a uniform definition on legacy system, in 1995, there was a definition proposed in [1]: if a system is important to the enterprise, and at the same time, the enterprise doesn't know how to update it to meet the new need, the system can be called as a legacy system. To legacy system, nowadays, there are three basic models to rebuild it to meet the new need.

Giving up legacy system: the model will give up a legacy system and a new system is developed to replace it at the same time [2]. The model will produce waste to the enterprise's resource, so it isn't used by most enterprise.

Wrapping legacy system: the model will hide the real implementation of legacy system, and its functions are published as public interfaces, then we can access the legacy system through these public interfaces [3]. In the model, the security and stability of legacy system aren't affected, and the rebuilding cost is low.

Rebuilding legacy system: in the model, before rebuilding the legacy system, we must analyze its functions and demands, and then we can implement its functions in the new system [4]. In fact, for the half-baked documents of legacy system, it is very difficult to analyze its functions.

In the three models, the second model is adaptive to enterprise. In the model, the inherent security and stability of legacy system are not affected in the new system, and this is very important to an enterprise.

The reminder of this paper is structured as in the following: section 2 gives some background information in legacy system integration technology, section 3 gives an introduction on the multi-agent based system encapsulation model. In section 4 gives a case study to describe how to encapsulate the Matlab as web components based on multi-agent technology, and finally, section 5 contains the conclusion.

Legacy System Integration Technology

XML Integration

The Extensible Markup Language (XML) is a broadly adopted

format for structured documents and data on the Web. XML is a simple and flexible text format derived from standard generalized markup language and developed by the World Wide Web Consortium. XML is expanding from its origin in document processing and becoming a solution for data integration. XML excels in inter-application data exchange because of its flexible and extensible method for describing data and its capability to communicate over the Internet using the standard HTTP protocol.

This flexibility makes XML a powerful mechanism for business-to-business (B2B) application integration. B2B integration is the automated exchange of information between systems from different organizations. B2B, for example, improves external processes such as supply chain integration or shipping/logistics tracking. XML-based B2B is gaining momentum as XML vocabularies emerge in specific business domains such as finance. In addition, a growing number of commercial enterprise application solutions are embracing XML [5].

Database Replication

Database replication is the process of copying and maintaining database objects in multiple databases that make up a distributed database system. Changes applied at one site are captured and stored locally before being forwarded and applied to the centralized repository. Replication provides users with fast, local access to shared data and greater availability to applications because alternative data access options exists. Even if one site becomes unavailable, users can continue to query, or even update, data at other locations. Database replication is often used to enable decentralized access to legacy data stored in mainframes. In this approach, local instances of modern databases are replicated parts of a centralized database. We can get a similar effect using a remote database gateway, but it is sometimes undesirable to use a software layer to wrap legacy data. Instead, we want a local copy of the data that wraps and buffers the original data source. New applications using the data receive the benefits of local access to a modern database

instead of the problems of remote access to an obsolete data repository.

Object-Oriented Wrapping

Objects have been used to implement complex software systems successfully. Object oriented systems can be designed and implemented in a way that closely resembles the business processes they model. Additionally, the use of abstraction, encapsulation, inheritance, and other object orientation (OO) techniques make object-oriented systems easier to understand.

To support object distribution, we need a more powerful means of communication than that provided by normal inter-object communication mechanisms. Distributed object technology (DOT) is the combination of distributed technology with OO. In effect, DOT extends object technology to the net-centric information systems of modern enterprises by using object middleware. The most prevalent object middleware is the Common Object Request Broker Architecture (CORBA) from OMG, with its platform-neutral object specification language, robust remote method calls, interoperable protocols, and rich set of services. The conceptual model of object-oriented wrapping is deceptively simple: individual applications are represented as objects; common services are represented as objects; and business data is represented as objects. In reality, object-oriented wrapping is far from simple and involves several tasks including code analysis, decomposition, and abstraction of the OO model [6].

Legacy System Encapsulation Model Based on Multiagent Technology

Based on multi-agent techniques, every function of legacy system can be encapsulated as a web component, which may have various interfaces for different functions. The multi-agent based encapsulation architecture is shown **Figure 1**.

External wrapper provides various interfaces to accept SOAP request, which can be managed by the client's power, and inner wrapper are developed for the legacy system in the institutions. The SOAP requests from online user are transmitted to external wrapper, and then retransmitted to inner wrapper. Then the inner wrapper decomposes the requests into a sequence of status values, according to which web component will change the state, and related web components are called to respond to the request. Finally the result should be sent back to external wrapper which in the same way sends it to the online user.

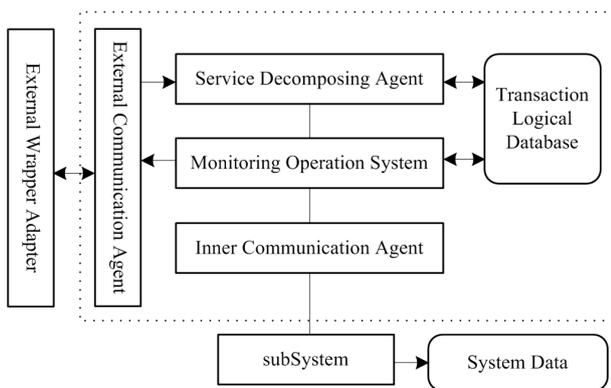


Figure 1. System encapsulation based on multi-agent technology.

External Wrapper

External wrapper accepts requests and cooperates with inner wrapper. Its main tasks are to analyze and decompose the requests, including creating service detailed lists, decomposing request to status switch, defining communications protocols and the format of input and output, registering and issuing web components in the UDDI.

External wrapper is closed to the actual operation platform, and access request through inner wrapper, external wrapper is used to establish and maintain the safe communication among different web components. When communication happens among different legacy system systems, the conversion between bytes and certain data type must be run to ensure platform-independent. The communication information is described and defined uniformly in XML.

Inner Wrapper

Inner wrapper needs to coordinate with external wrapper, and interact with legacy system systems directly. Therefore, they should be standardized and adaptable to the operation environments of legacy systems. The architecture of inner wrapper is shown in **Figure 2**.

The net communication agent receives service request from external wrapper, and sends it to service decomposing agent, which decomposes the service request to the status switch set based on transaction logic library. The net communication agent is composed of external communication agent and inner communication agent. Then according to the status data, the monitoring operation agent manipulates related subsystems and sends the needed output to external communication agent.

Based on multi-agent technology, the legacy system is encapsulated with common interfaces, and other systems can communicate information with the legacy system based on the common interfaces, but to the clients who don't be familiar with the common interfaces, we can't send the request with right format to the legacy system and get the right result.

For providing the access interface to clients, as shown in **Figure 3**, an encapsulation architecture of legacy system is proposed, as following describes how to clients to send the request to the legacy system and get the return results.

- Clients send the request message to the application message with SOAP format.

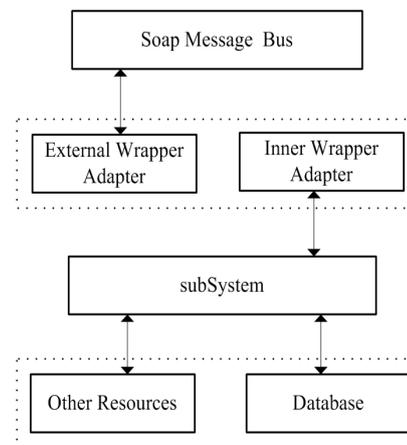


Figure 2. The architecture of inner wrapper.

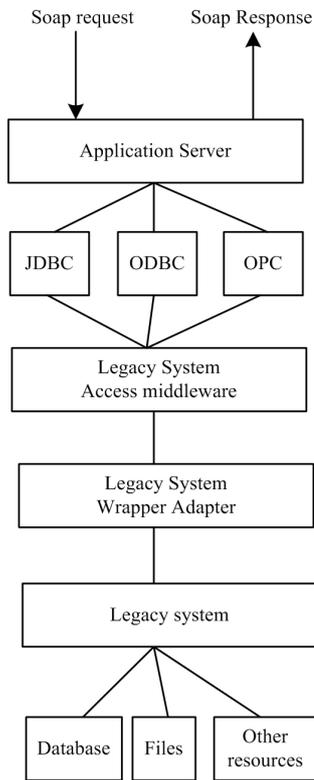


Figure 3.
The encapsulation architecture of legacy system.

- Application server receives the request and sends it to the legacy system access middleware based on the JDBC, ODBC or OPC interface.
- The legacy system access middleware sends the request to the legacy system wrapper adapter, which includes external wrapper and inner wrapper.
- The legacy system wrapper adapter sends the request to the legacy system based on their common access interfaces.
- The legacy system receives the request, and returns the result to the legacy system adapter, and finally, the result is returned to the clients.

Based on the proposed encapsulation architecture, the legacy system is encapsulated as common web components, and then we can integrate the legacy system with other legacy systems, and these legacy systems can share message with SOAP format. Due to adoption of wrapping technology for legacy system, original security and stability of legacy system are guaranteed in the web components.

Case Study

Matlab is professional software developed by MathWorks Corporation. It has powerful function on data analysis, data emendation, and image process etc. Based on the proposed system encapsulation model, we can encapsulate it as Web component to provide the computation functions, linear regression analysis, nonlinear regression analysis, matrix normalization, covariance matrix computation [7,8]. Regression analysis is mainly use to analyze real-time data for the product devel-

opment process.

As shown in **Figure 4**, the Matlab is encapsulated as common web components, as shown in **Figure 5**, the SOAP message can be sent to the Matlab common component to request linear regression function, and then the result will return with SOAP message as shown in **Figure 6**.

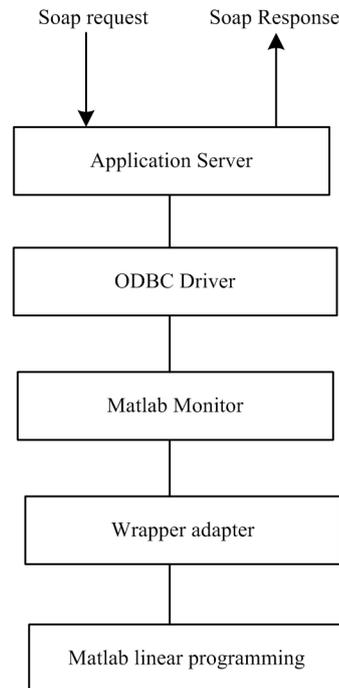


Figure 4.
The encapsulation model of Matlab.

```
POST /WebService1/Matlab_Service.asmx HTTP/1.1
Host: 202.38.218.217
Content-Type: text/xml; charset=utf-8
Content-Length: 524
SOAPAction:"http://tempuri.org/WebService1/Matlab_Service/Line_Regression_Analysis"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Line_Analysis xmlns="http://tempuri.org/WebService1/Matlab_Service">
<DateFrom>2009-3-11 10:49:38</DateFrom>
<DateEnd>2009-3-11 14:32:11</DateEnd>
</Line_Analysis>
</soap:Body>
</soap:Envelope>
```

Figure 5.
SOAP request message of linear regression service.

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 487
<?xml version="1.0" encoding="utf-8">
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  < Line_AnalysisResponse
    xmlns="http://tempuri.org/WebService1/Matlab_Service">
    < Line_AnalysisResult>
      Y=0.010*x1 + 0.08*x2 + 0.001 * x3 + 90*x4 + 35.529*x5-
        0.019*x6
    </ Line_AnalysisResult>
  </ Line_AnalysisResponse>
</soap:Body>
</soap:Envelope>

```

Figure 6.
SOAP response message of linear regression service.

Conclusion

For encapsulating the enterprise legacy system as common web components, a multi-agent based legacy system encapsula-

tion model is proposed. Due to adoption of wrapping technology for legacy system, original security and stability of legacy system are guaranteed in the web components. Finally, the validity and practicability of the migration technology are verified through the application in the encapsulation of Matlab as web components.

REFERENCES

- W. Proie, "Legacy computer systems migration," AISTech - Iron and Steel Technology Conference Proceedings, 2, pp.873-881, 2009.
- Y. Qu, "Research and application of reusing enterprise legacy systems based on Web Service," 2009 WRI World Congress on Software Engineering, 3, pp.325-329, 2009.
- X. Li, "A service-oriented partitioning and encapsulation method for java legacy systems," Jisuanji Xuebao/Chinese Journal of Computers, 32(9), pp.1804-1815, 2009.
- W. Gee, "Systems engineering requirements for legacy hardware upgrade and sustainment requirements definition, analysis, and validation," 2009 IEEE International Systems Conference Proceedings, pp. 431-436, 2009.
- B. Medjahed and A. Bouguettaya, "Composing web components on the semantic web," Journal on Very Large Databases, 4 , p. 333-351(2003).
- J. Y. Sheng and J. Hang, "CORBA based tele-collaboration workflow mode," Computer Applications, 9, p. 19-25(2002).
- Y. Zhan, J. A. Clark, "A search-based framework for automatic testing of MATLAB/Simulink models," The Journal of Systems and Software, 81(2), pp.262-285, 2008.
- C. Y. Chen, Y. C. Shun, "MATLAB-based rapid controller development platform for control applications," Journal of Mechanical Engineering Science, pp.1461-1473, 2007.