AN OPEN WEB SERVICE-BASED DSS

Si Yaqing[1], Chen Yonggang[2] and Zhang Shaofeng[3]
1 Economics and Management School of Beijing University of Posts and Telecommunications, No.10, Xi Tu Cheng Road, Haidian District, 1000876 Beijing, China, siyaqing@vip.sina.com.cn
2 Economics and Management School of Beijing University of Posts and Telecommunications, No.10, Xi Tu Cheng Road, Haidian District, 1000876 Beijing, China, tochenyg@gmail.com
3 Economics and Management School of Beijing University of Posts and Telecommunications, No.10, Xi Tu Cheng Road, Haidian District, 1000876 Beijing, China, zhshf@263.net

Abstract. This paper presents a new and open Web Service-Based DSS framework under distributed service-oriented architecture. In the past, DSS always refers to a software product. But in this new framework, DSS appears as a website, we called it as decision support website. This paper explores the details of the architecture of the decision support website and describes the business mode about this website. Then we give the whole implementation process of web service. The new DSS change the features of the traditional DSS completely by adopting web service application integration technology and introducing into new business mode.

1 Introduction

Decision Support System (DSS) is an application software system, which is used to help decision maker to solve half structured or unstructured problems by using models, data and knowledge [1]. Traditional DSS is composed by user interface component, data base, model base and their management system. Model base is the primary component of DSS. With the rapid development of computer and network technology, the development of DSS application software system has experienced various different phases. Especially with the emergence of Web 2.0 concept, new DSS appears unceasingly.

A Web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging [2]. It covers all the details necessary to interact with the service, including message formats (that detail the

Please use the following format when citing this chapter:

Yaqing, S., Yonggang, C., Shaofeng, Z., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.609-614.

operations), transport protocols and location. The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written. This allows and encourages Web Services-based applications to be loosely coupled, component-oriented, cross-technology implementations.

In this paper, we present a new DSS architecture that is open and based on Web Service. Unlike the traditional DSS which generally refers to a software product, in this architecture, the DSS appears as a website on the Internet. We called this new DSS as decision support website. Every model and data is encapsulated as independent Web Services. DSS makes decision by calling these model and data web services. Meanwhile, we found a private UDDI registration center, by registering in this UDDI registration center, everyone in the world can add model and data into this open DSS.

2 The Architecture of Web Service-Based DSS

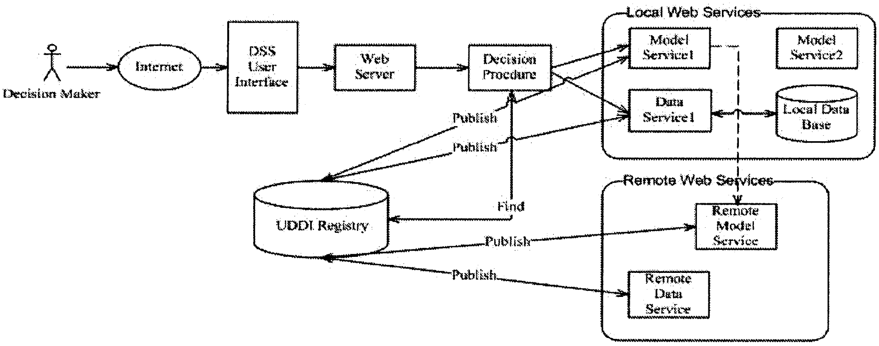

Fig. 1. The Architecture of Web Service-Based DSS

Fig. 1 illustrates a new DSS architecture that is based on web service. In the technology framework, this new DSS uses the web service technology which is a distributed calculating technology. DSS makes decisions for people by invoking a series of web service. These services may belong to the developer of DSS and other corporations as well. This DSS consists of following components: Web Server, UDDI Server, Web Services and local Database. The architecture of this DSS can be divided into three layers. From top to down, they are Web Services Layer, Web Server Layer and User Interaction Layer (Browser).

1) Web Services Layer

Web services layer includes a set of web services. In this new architecture, the model of traditional DSS is encapsulated as an independent web service. It can be used alone or with other Web Services to carry out a complex aggregation or a business transaction. There are two kinds of web services: local web service and remote service. Local web service is an application unit that developed by the DSS developer. Remote web service is developed by other corporations or individual. DSS does not know how these services implement, which programming language is used to develop these services. It only knows the function of each service. After publishing in the UDDI registry center, DSS can invoke these services by Internet. In the traditional DSS, DSS application interacts with database directly. Bu in this new technology architecture, we add a data service between DSS application and database. When DSS want to acquire data from database, it needs to invoke this data service to interact with database indirectly. By adding a data service between DSS and database, this new DSS becomes more independent and general.

2) Web Server Layer

Like the Web Server in the traditional DSS based on B/S, in this architecture, DSS application and local web services are all deployed on Web Server.

3) User Interaction Layer

The main component of user interaction layer is web browser. Users log on the DSS by browser, and interact with DSS. After a series calculation, web server sends results to the user's browser. Web browser is only a tool used to display information and results. Using browser to interact with DSS makes the DSS becoming platform-independents.

To meet all kinds of demands which may come from different industry and different people, DSS needs to have all kinds of models and data. There are hundreds and thousands of models, and they are impossible to be developed by one corporation. So this DSS is designed as an open application system. In this open DSS, other corporations and peoples can add their models into DSS with web service form. UDDI registry center just likes a bridge between DSS and other service providers. By registering on the UDDI center, DSS can invoke these remote web services on the Internet. By integrating other web services into this open DSS, it can become a comprehensive decision support system.

We also present a new business mode that is very different from traditional mode. In the past, DSS always is a software product that developed for one enterprise. In this new architecture, the DSS is not a product again, it appears as a website, and we called it as decision support website. Every people in the world can logon this website to help them to make decisions. The user of website should pay fees to website operator. To encourage other enterprises or peoples to add their model services into website, website also pays fees to service provider. In this new architecture, the main tasks of software develop corporation is operating the decision support website.

3 The Implementation of Web Service

3.1 Developing a New Web Service

The main tasks of this phase are developing and testing a web service. There are many ways to develop a web service. You can develop a new web service completely or transfer an existing application to a web service, or you can compose other web services to become a bigger web service. There are many platform that can implement web service, we use Microsoft's .NET platform to establish a new web service which can calculate the area of a circle [3].

```
<%@ WebService Language="C#" Class=" CalculateArea" %>
using System;
using System.Web.Services;
public class CalculateArea : System.Web.Services.WebService
{
  [WebMethod]  public double Circle (double r)
  {
    double t = Math.PI * r * r;
    return t;
  }
}
```

Inputting the above codes into any text edit tools, saving the document under the directory of "WWW ROOT" in Web Server, renaming the document as "Area.asmx", now we have established a web service. We can test this web service by inputting "http://localhost/Area.asmx" into browser's address bar. In the above document, it defined a "CalculateArea" class that inherited from "WebService". In the "CalculateArea" class, it defined a public method named "Circle". This method has one input parameter whose type is "double" and returns the area of a circle depending on the input parameter which indicates the radius of a circle.

3.2 Producing WSDL Document

Web service description document has an important role in the web services architecture. It describes the function and location of a web service, the type of transfer protocols and the interface of operations and so on. It is through the service description that the service provider communicates all the specification for invoking the web service to the service requestor [4].

Web services architecture uses WSDL for service description. WSDL is an XML document that can be created by tools automatically or produced manually by inputting XML tag into any text edit tool. For instance, the description of above web service which calculates the area of a circle can be acquired by "http://localhost/Area.asmx?wsdl" URL.

3.3 Publishing a Web Service

The purpose of publishing a service is making service requestors can discover and invoke this service. A service description can be published using a variety of mechanisms. These various mechanisms provide different capabilities depending on how dynamic the application using the service is intended to be [2].

The simplest publishing mechanism is the direct publish. A direct publish means the service provider sends the service description directly to the service requestor. This can be accomplished using an e-mail attachment, an FTP site [2]. Direct publish can occur after two business partners have agreed on terms of doing business over the web, or after fees have been paid by the service requestor for access to the service. In the above DSS framework presented in this paper, if another corporation which has lots of model services want to add their model services to this open DSS, it can use a direct publish mechanism. In this case, decision support website can maintain a local copy of the service description for invoking. Another publishing mechanism is that service provider publishes web services on UDDI registry center. This mechanism may suit to individual who has a little web services and want to add these services into decision support website.

3.4 Web Service Discovery and Binding

How to discover a web service depends on the style of web service publishing. If a web service is published directly to service requestor. In this situation, service requestors can get WSDL document directly from service provider, then they can make use of service description to create right SOAP invoke message. If a web service is published on UDDI registry center, service requestors can search a web service by interacting with the interface supported by the UDDI node.

In the above DSS framework presented in this paper, DSS will invoke web services during two different phases of an application lifecycle-design time and runtime. At design time, DSS creates an agent class according the service descriptions. Then DSS can use this agent class to invoke the method of a web service just like local class [5]. At runtime, DSS queries service descriptions dynamically, then according to the descriptions, DSS creates right SOAP request message, and receives SOAP response from remote machine.

4 Conclusions

In this paper, we present a new DSS architecture based on web service. It is very different from traditional DSS both in technology and business mode as well. We give the whole implementation process of web service.

References

1. G. HongShen, *The Theory, Method and Case of DSS* (TsingHua University Press, Beijing, 2005).
2. H. Kreger, Web Services Conceptual Architecture (WSCA 1.0), IBM Software Group, 2001.
3. Z. XiaoPing, *The Kernel of .NET – The Theory and Development of Web Service*, (People Posts and Telecommunications Press, Beijing, 2002).
4. J. Colgrave and K. Januszewski, Using WSDL in a UDDI Registry, http: / / www. w3. org
5. Z. ZhaoQuan, X. YongSheng, and H. ZhengHua, The Study of Model Reuse Mechanism in DSS Based on Web Service, *Journal of Xiamen University (Natural Science)* **43**(6), 2004.