A Realization of the Dynamic Procurement Web Service in Practical Use Case

Takashi KOSHIDA  
Matsue National College of Technology  
14-4 Nishi-Ikuma, Matsue, Shimane 690-8518 Japan  
takashi@matsue-ct.ac.jp

Takashi KANEYAMA  
Faculty of Information, Shizuoka University  
3-5-1 Jyohoku, Hamamatsu, Shizuoka 432-8011 Japan  
cs1502@s.cs.inf.shizuoka.ac.jp

Kenji HATANO  
Nara Institute of Science and Technology  
8916-5 Takayama, Ikoma, Nara 630-0192 Japan  
hatano@is.aist-nara.ac.jp

Shunsuke UEMURA  
Nara Institute of Science and Technology  
8916-5 Takayama, Ikoma, Nara 630-0192 Japan  
uemura@is.aist-nara.ac.jp

Abstract

We developed the dynamic procurement Web Service by using AXIS-SOAP, UDDI and WSDL technology. We researched and confirmed the problem of practical use on the UDDI registry. We assume a wholesaler, retailers, manufacturers and the grocery association in our use case. The retailer gives an order to the wholesaler. The wholesaler confirms and quotes for the manufacturer for requirement. If there is an inventory shortage, the wholesaler dynamically retrieves another manufacturers by using the UDDI registry and complements for the shortage. We researched and developed these business processes in two different ways. We present the defect of the discovery of Web Service using the UDDI registry and propose a semantic modification of UDDI.

Keywords: SOAP, UDDI, WSDL, Web Service, Semantic Web, Use case

1. Introduction

As spreading the Internet, CALS (Continuous Acquisition and Life cycle Support; Commerce At Light Speed), which is to do electronic data processing in the enterprise activities widely, progressed in various industrial fields. Although B2B (Business-to-Business) system nowadays diffuses globally, it is limited to the regular transactions between the specific enterprises. But the XML-based architecture, SOAP (Simple Object Access Protocol)[1], UDDI (Universal Description, Discovery and Integration) [2] and WSDL (Web Services Description Language)[3], are enabled to find and use dynamically the world-widely business transaction. Now these are called Web Services. SOAP is a XML-based protocol intended for exchanging structured information in a distributed environment. UDDI is a web-based framework that registers, retrieves and connects information about a business or other entity and Web services. UDDI data structure consists of four parts (i.e. “businessEntity”, “businessService”, “bindingTemplate” and “tModel” data structures). The “businessEntity” describes information about a enterprise that provides Web Services. The “businessService” describes information about Web services. The “bindingTemplate” describes technical information about Web Services. These three data construct a nested structure. The “tModel” describes specification of Web Services. WSDL is a XML-based language. WSDL provides a uniform way of describing the abstract interface and arbitrary protocol bindings of Web Services.

When we use Web service, at first, we must discover Web service by retrieving the UDDI registry. But it is so difficult to find easily and rapidly Web service that we need. The IBM UDDI registry provides the GUI-based retrieval system. But in this system, we retrieve only by using “business name”, “service name” or “tModel name” in each industrial category and category code like NAICS, UNSPSC, and so on. And we must select Web service by referring to the information displayed on the web.

In this paper, we assume the procurement system dealing with goods distribution among a wholesaler, retailers and goods manufacturers as a practical use case [4]. We propose and study a new one more database, which helps for dynamically to discover Web services. We assume that the grocery association
manages the database and helps the wholesaler to find manufacturers and Web services. We researched and developed this procurement system under Windows XP, Java Development Kit(Jdk)1.4.0_01[5], Apache-Axis 1.0 [6] and Tomcat 4.0.6 [7]. We registered our Web services to the IBM UDDI Test Node[8] and experimented the business flow in practice.

2. System model
In this Section, we will present our B2B system model by providing the following use case: we assume the procurement system dealing with goods distribution among a wholesaler, retailers and goods manufacturers. Retailer gives an order to wholesaler. The wholesaler receives orders from many retailers and confirms the stock of the goods ordered for the manufacturers, which carries on transactions regularly. If the manufacturer has stocks, wholesaler purchases an order immediately. If the product is out of stock or new goods, the wholesaler dynamically refers to another manufacturer by retrieving the UDDI registry. After retrieval, the wholesaler confirms the stock and estimates the price of goods for manufacturers by using Web services that manufacturers provide. If there is sufficient stock and the price of goods is reasonable, then the wholesaler selects the trustworthy manufacturer among these and gives an order. In practice, the wholesaler accepts several orders from many retailers to different manufacturers and must delivery punctually the goods ordered at each retailer. It is necessary for the wholesaler to realize fully the delivery rate of goods and to inform retailers of the delivery status of goods. Further, it is required for retailers to confirm the delivery state of goods in any time.

As we described in Section 1(i.e., 1.Introduction), the most difficult step in these business processes is dynamically to find other manufacturers that provide the sold out goods or new order goods and to execute Web service that manufacturers provide. Of course, before executing Web service, we must know how to use Web service. That is, how many parameters are there and what are a concrete parameter value and so on. So, for the purpose of supporting this process, we introduce a new database and assume the grocery association that manages the database. The grocery association provides the manufacturer’s information to help wholesaler to retrieve the UDDI registry and to execute manufacturer’s Web service. We call such a complementary database as the “registrar”.

And we compared with the two different business process in the following approach.

(1) The wholesaler finds manufacturers and WSDL’s URL of Web service directly retrieving the IBM UDDI registry by using programs that we developed.

(2) The wholesaler uses the grocery association’s database to find manufacturers and to execute Web service that the manufacturers provide.
We show the previous (2) business process in Figure. 1. The number (e.g., 1., 2..) indicates an order of execution.

3. System function and Database structure
In this section we describe the system function and database structure in the wholesaler, manufacturer and the grocery association.

3.1 Wholesaler
We developed the wholesaler’s system by using Java Server pages (JSP). We describe the database structure, system function and business flow in the wholesaler.

(1) Database structure
Wholesaler has a database containing three tables. One is “Order” table, the others are “Manufacturer” and “WebService” table. “Order” table has the next fields: voucher number, retailer ID, JAN code (means product ID), goods name, request quantity, ordered delivery day, ordered delivery shop, ordered time, completion code. “Manufacturer” table has the next fields: JAN code, goods name, manufacturer’s name, Web service name which the manufacturer provides, number of Web service’s method and the access point of Web service. “WebService” table has the next fields: manufacturer’s name, Web service name that the manufacturer provides, and method name of Web service. This “Manufacturer” table lists the
manufacturer that transacts regularly, so the wholesaler already knows about the Web services, which the manufacturer provides.

(2) System function and business flow
Retailer gives an order by accessing Java server pages in the wholesaler. The wholesaler stores the order to the “order” table. Next, the wholesaler retrieves the “Manufacturer” table using “JAN code” and if the manufacturer which corresponds to the “JAN code” is found, the wholesaler confirms the stock by using manufacturer’s Web service (we developed the “PriceAndAvailable” Web service method). The “PriceAndAvailable” method returns the available stock to the wholesaler, so the wholesaler replies the current status of order to the retailer as a mark. Mark “circle” means that there is sufficient goods of stock and already is ordered. Mark “triangle” means that there is not enough. Mark “cross” means that there is nothing or does not order yet.

If the product is out of stock or insufficient, the wholesaler must find the manufacturer that supplies the goods. Next step is the retrieval of another manufacturer that provides the insufficient or shortage goods. This is an important step of this procurement system.

In this case, we compared with the two different retrieving approaches and practiced.

a. The wholesaler directly retrieves a WSDL’s URL of Web service from the IBM UDDI registry.
The wholesaler retrieves directly the IBM UDDI registry. We developed the several retrieval programs by using UDDI4J [9]. We can retrieve by using “name” (means “business name”) and “description” in the “businessEntity” tag. In this case, to find manufacturer, it is necessary that “business name” represents the name of goods species, and the goods name that manufacturers produces must be described in “description” tag. If not, the wholesaler cannot find manufacturer.

We can get the “businessKey” number that identifies as each enterprise. The “businessKey” number is attached automatically to the “businessEntity” tag when an enterprise registers Web service, and is a number that consist of unique 128 bit digits.

Next we can find the “serviceKey” number in the “businessService” tag by using the “businessKey” number and we can find the “bindingTemplate” tag by using the “serviceKey” number. Also we can get the “accessPoint” of Web service from the “bindingTemplate” tag. There is a “tModelKey” number in the “tModelInstanceInfo” in the “bindingTemplate” tag. Finally, we can find the “tModel” tag by using the “tModelKey” number. We analyze the “tModel” tag and we get a WSDL’s URL from the “overviewURL”. WSDL file describes Web service and how to access them. We can create automatically the stub program in client side by using WSDL2Java. Further, by using WSIF [10] we developed the stub-less client program that executes automatically from retrieving a WSDL’s URL until executing Web service.

For example, in the case of receiving the order of rare wine, it is necessary that there is a description about the rare wine (i.e. wine name, production year and region, feature and so on) in the “description” tag of “businessEntity” tag. And it is necessary that the word “wine” is included in the “name” tag of “businessEntity” tag. Until the wholesaler finds manufacturers which product the shortage goods, the wholesaler repeats this retrieval.

b. The wholesaler is with the assistance of the grocery association’s registrar.
The wholesaler uses Web service of the grocery association. We developed the “RegistrarService” Web service (including two methods: “findMakerByProductCode” and “getMakerDetail”). About this Web service, we describe in section “3.3 Grocery association” in detail.

3.2 Manufacturer
(1) Database structure
Manufacturer has a database containing three tables. One is “ReceivingOrder” table; the others are “Stock” and “Wholesaler” table. “ReceivingOrder” table has the next fields: voucher number, ordered
goods JAN code, ordered goods name, price per item, ordered goods quantity, total price, ordered delivery day and real delivery day. “Stock” table has the next fields: goods JAN code, goods name, stock quantity, and price per item. And “Wholesaler” table has the next fields: voucher number, wholesaler ID, wholesaler name and receiving order day. Even when the same wholesaler gives orders of some goods, “Wholesaler” table has only one record in each wholesaler.

(2) System function and business flow
In our use case, we assume the manufacturer that provides a Web service named “MakerService”. “MakerService” provides two methods. One is “PriceAndAvailable” method and the other is “RequestOrder” method. The manufacturer provides this Web service for the wholesaler.

a. Stock/quote service (Web service method: PriceAndAvailable)
“PriceAndAvailable” method returns an available stock value and quote in opposition to JAN code, request quantity, and delivery day.

b. Receiving order service (Web service method: RequestOrder)
After confirming the available stock and quote by using “PriceAndAvailable” method, wholesaler gives an order by using this method. “RequestOrder” method deals with the purchase order (JAN code, goods order quantity, delivery day) from wholesaler. The “ReceivingOrder” table and “Wholesaler” table are created newly; the “Stock” table is updated.

3.3 Grocery association
The grocery association provides the “RegistrarService” Web service for the wholesaler, which are corporate with the UDDI registry. The “RegistrarService” has two methods, “findMakerByProductCode” and “getMakerDetail”. The grocery association provides the manufacture’s information for the wholesaler. Because when there is an inventory shortage for the retailer’s order, it is so difficult that the wholesaler immediately finds the manufacturer and executes the manufacturer’s Web service by retrieving the UDDI registry. We assume that the grocery association manages the database as describing below by retrieving and searching the UDDI registry.

(1) Database structure
The grocery association has a database containing two tables. One is “Product” table and the other is “MakerInfo” table. “Product” table has the next fields: goods classification code and goods classification name. For example, goods classification code is “1111”, classification name is “wine”. “MakerInfo” table stores the manufacturer’s information and has the next fields: goods classification code, businessKey, manufacturer’s name, business contents, product names, capital, name of main bank, stockholders and main customers. The “MakerInfo” table restores the “businessKey”, manufacturer’s name, and other important information (i.e. capital, stockholders, main bank) that is not registered the UDDI registry. This information is helpful to the wholesaler in finding out the trustworthy manufacturer.

(2) System function and business flow
When there is an inventory shortage for the regularly transacting manufacturer the wholesaler must retrieve another manufacturers. The “RegistrarService” is used for finding out manufacturers by the wholesaler. The manufacturers that produce the shortage goods are retrieved as below.

a. Manufacturer retrieval service (Web service method: findMakerByProductCode)
“Product” table is used for the retrieval of the UDDI registry. The wholesaler designates the “goods classification name” of the shortage goods, for example, if coffee is short of, the corresponding “goods classification code” is “4444”. And this “goods classification code” is given to the Web service method “findMakerByProductCode”, this method retrieves “MakerInfo” table by using the code “4444”, and returns the “businessKey”, manufacturer’s name. Furthermore, this method retrieves the UDDI registry
by using the “businessKey”, and finally returns the “accessPoint”, WSDL’s URL address of the manufacturer’s Web service in “bindingTemplate” tag. All of manufacturers that belong to the same “goods classification code” are represented to the wholesaler.

b. Manufacturer’s information service (Web service method: getMakerDetail)

This method represents the detailed information about manufacturers. After executing the Web service method “findMakerByProductCode”, the wholesaler can find the information (i.e. “businessKey”, “manufacturer’s name”, “accessPoint”, “WSDL’s URL”) of some manufacturers that produce the shortage goods. Next, the wholesaler selects the one manufacturer’s “businessKey”, so this “getMakerDetail” method shows the manufacturer’s information (i.e., manufacturer’s name, business contents, production goods, capital, stockholders, main bank, main customer) by retrieving the “MakerInfo” table, and represents the “personName”, “phone”, “email”, and “address” by retrieving the “contact” tag in the UDDI registry.

By referring this information, the wholesaler can select the trustworthy manufacturer and can execute the transaction of compensating the shortage goods for the retailer’s requirement. So the wholesaler is able to dynamically retrieve the manufacturer when there is an inventory shortage.

4. Conclusion

We believe that for constructing the useful system we should apply concretely the new technology to use case. So we assumed the procurement system, which consisted of retailer, wholesaler, manufacturers and grocery association. We developed the dynamic B2B system using Web services and explained about that. Through our research and development, we believe that the UDDI registry is the most core and important architecture in Web service’s technology. But in practice, we think it is difficult that we quickly discover the necessary Web services from the UDDI registry. It is most important that we can easily discover Web services, which we really need. So in this time, we introduced the grocery association, which provides the information to complement the UDDI registry. WSIL [11] is a standard like UDDI, but WSIL intends to discover a distributed Web service not registered UDDI. From a point of view that unifies the information about discovery and execution of Web service, it is not so good that the other registrar is in existence like this grocery association’s database. So, to overcome this defect, we propose to construct the semantic UDDI registry. If so, agents can discover Web service and other information that user needs.

The use of agents in E-Commerce field is surveyed [17]. Nowadays it is proposed some standards for composing Web service like WSFL [12], XLANG [13], BPEL 4WS[14], WSCI [15] and DAML-S [16] and research papers in the field published [18, 19]. Semantic web using DAML-S is proposed and researched [20, 21], but we think that it is not yet in practical use. So we continue to research and develop semantically Web service as attaching importance to UDDI and WSDL.

References