Article ID: 1003－1251(2014)06－0087－05

The Research and Realization of UDDI in Service-Oriented Architecture

ZANG Tianxiang[1],TIAN Minghao[1],ZHANG Liyuan[2],FENG Yongxin[1]

(1. Shenyang Ligong University,Shenyang 110159,China;2. Renmin University of China,Beijing 100872,China)

Abstract: It is a trend that SOA is becoming the future enterprise software architecture world widely. The key point is how to describe and discover services,thus these mainly rely on UDDI. Firstly,the current situation of SOA and UDDI are introduced,and then UDDI V3 and related technical specifications are analyzed. Secondly,UDDI models,working principle,the mapping of WSDL and UDDI,and its implementation technologies are delved into. On this basis,the overall framework of UDDI is built and the service publishing and inquiring process are analyzed. Finally, a registry based on the latest standard of UDDI is achieved.

Key words: SOA; UDDI; JUDDI; SOAP; Web Services

CLC number: TP393.04 **Document code:** A

SOA(Service-Oriented Architecture) which is coarse-grained, loosely coupled software architecture,communicates by a simple and precise definition of the interface between services,while the underlying programming interfaces and communication model are not involved. The coupling between the systems could be greatly reduced by using the SOA framework,so the reusability of software components will be increased. In the SOA architecture, UDDI(Universal Description Discovery and Integration) is the core of the whole architecture and acts as a bridge between service providers and consumers[1]. UDDI is Web-based, distributed, providing registration center standards for the web services,also contains a set of standards enable other companies can find services which is published. Nowadays,UDDI version 3.0.2 which is the latest has been approved as the standards for the structured Information and promotion organization. Compared to V1,V2 version,UDDI V3.0.2 solves the important problems in developing the web service,such as improved the safety and Interoperability between UDDI and being more international[2]. Microsoft and IBM and some other IT giants have closed the public registry, as they are optimistic about UDDI V3. In China, did not follow them quickly to strengthen the research on new standards based on UDDI V3.0.2. The UDDI based on latest technology an integral part in the SOA,so it´s meaningful to have more research on this field.

1 THE ANALYSIS OF UDDI V3

UDDI V3 proposed a new UDDI system framework that there will be a lot of different UDDI registries, and every particular UDDI registry is composed of one or more nodes,and a huge system will be formed via these UDDI[3]. UDDI specifica-

Received date: 2013－12－06

Biography: ZANG Tianxiang(1989—),male.

tion includes XML Schema of SOAP messages and API descriptions of UDDI, which establishes the basic information model and interaction framework. All APIs in UDDI are defined with XML, wrapped in a SOAP envelope, using the SOAP request and SOAP response models, transporting in the HTTP. Comparing the UDDI registry with searching engine, the searching engine returns a URL of web page only, while the UDDI registry not only returns the location of services, but also needs to return the information of services related, working method, parameters used and values of the services etc.

1.1 UDDI Data Structure

The core information model used by a UDDI registry is expressed in several XML schemas. XML was chosen because it offers a platform-neutral view of data and allows hierarchical relationships to be described in a natural way[4]. XSD was chosen because of its support for rich data types and its ability to easily describe and validate information based on information models represented in schemas. The UDDI XSDS define several core types of information that provide the kinds of information that users and applications would need to know in order to use a particular Web service. Together, these form a base information model and interaction framework of UDDI registries. They are: businessService、businessService、tModel、bindingTemplate、publisherAssertion、Subscription[5]. These, like all UDDI data types, are expressed in XML and are stored persistently by a UDDI registry. Within a UDDI registry, each core data structure is assigned a unique identifier according to a standard schema. This identifier is referred to as a UDDI key.

1.2 WSDL and UDDI

A complete WSDL(Web Services Description Language) is composed of a service interface and a service implementation document[6]. Service interface represents reusable service definitions, published as tModel in UDDI registry. And some elements of tModel are formed via the information of WSDL. Every service implementation document was defined by a WSDL service element; every element of the service implementation document was released as businessService. BusinessService element could be formed according to WSDL service implementation document. When publishing a WSDL service description, the service interface and service implementation can be published separately. For example, publish a service interface as a tModel and publish the service implementation as businessService. Benefits of this design, the interface definition and the service implementation can be located on different sites. The mapping from WSDL and UDDI is shown in Figure 1.

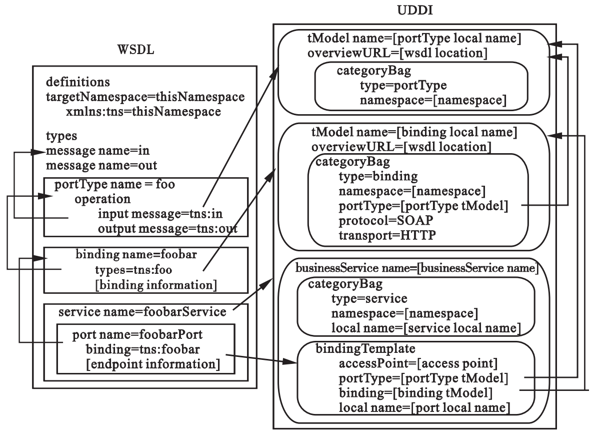


Figure 1 The mapping from WSDL to UDDI

The mapping describes a methodology that a WSDL1.1 defines to UDDI V3 data model. This methodology maps every WSDL artifact to a separate UDDI entity, and expresses the design of "building block" which WSDL described. wsdl: portType and wsdl: binding elements maps to uddi: tModel entity; wsdl: service element maps to uddi: businessService entity; and wsdl: port element maps to uddi: bindingTemplate entity. KeyedReferences provides a mechanism to represent additional metadata and the relation between the two UDDI entities.

2 THE OVERALL PROJECT DESIGN

JUDDI which is fully functional and simple to use, is UDDI registry server that meets the UDDI

V3 specification. JUDDI has many query interfaces and provides notification and copy services. JUDDI can use a variety of databases, such as mysql, oracle, postgresql, etc. All the tables and table structures in the JUDDI database are from the UDDI V3 specification. JUDDI established a separate table for the six kinds of data structures defined by UDDI V3 and all the information of every data structure. Here, to generate JUDDI V3 data structure by using the Navicat for Mysql, totally 46 database tables, a brief registry database model is shown in Figure 2.

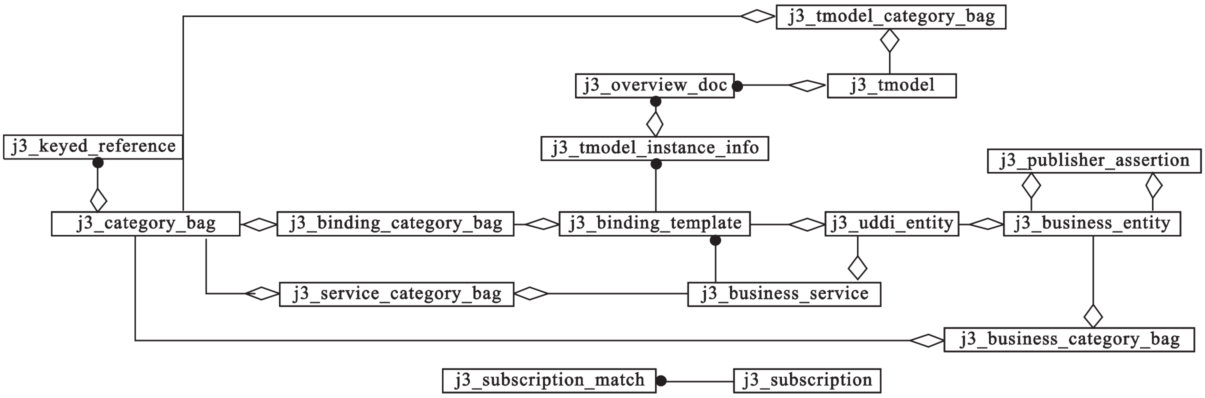


Figure 2　Brief model of JUDDI

According to the multi-tier architecture of J2EE and the function requirements in UDDI specification, the basic framework of the UDDI registry could be designed as shown in Figure 3. JSP template presents the page to the users and provides interaction with users. Control layer which receives, processes, sends data and control the flow, uses Struts2 framework to connect DAO layer and view layers. Business layer using spring to manage the entire structure transparently, providing IOC container to make the code loose couple and section function of AOP framework etc. JUDDI V3 server based on UDDI V3. 0. 2 specification mainly provides the Web and API interface. The core function is to define the data structure and discover and publish which are based on this data structure. UDDI data is stored in data storage layer. UDDI registry client interact with the JUDDI server through the SOAP, providing service publish for the service provider and offering the discovery for the service consumers.

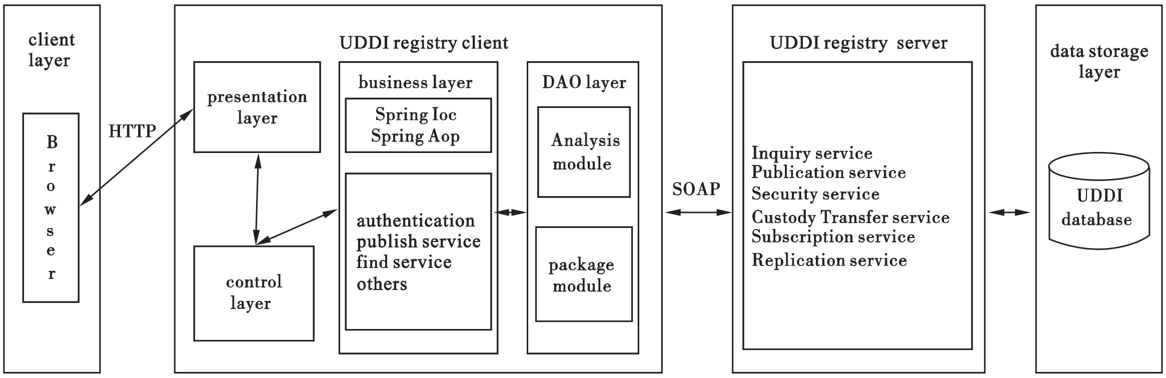


Figure 3　General framework of UDDI registry

3 THE IMPLEMENTATION OF UDDI REGISTRY

3. 1 The process of service query

Firstly, users send the information(user name and password) to the UDDI registry server through the UDDI registry client. Secondly, the server will search the user from the back-end database, get the user´s permission, and assign an identity token. Identity token is a UUID(Unique Universal Identifier) type data, which is stored in the database, can identify the user and associated with user information. Identity token is time-limited; they could no longer be used if expired. The copy is stored in the user client, and it´s needed to pass an identity token, when finding APIs are called. The service consumers build SOAP messages and send to the registry server. Finally, the server will call finding APIs and search the appropriate service information in the background data source, when receiving a finding request. Process of finding services is shown in Figure 4.

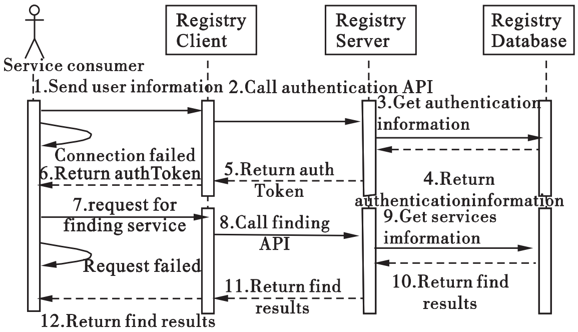


Figure 4 Process of finding services

3. 2 The process of service query

Businesses Registrations need to publish some useful information in UDDI registry, such as business and service description. UDDI registry assigns a programmatically unique identifier(UUID) which is associated with an entity. The UUID is valid only in the registry where it´s created. An authToken with the appropriate permissions is needed when a user publish services. Then, the authToken was required, when call publishing API. The publisher structures SOAP messages and send to UDDI registry server. When a publishing request is received, the server will call publishing API in order to save the service information into a UDDI database. Finally, the server will feed back the update results to publisher. The process of publishing services is shown in Figure 5.

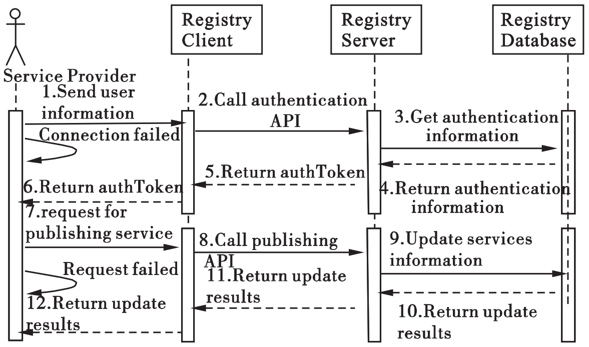
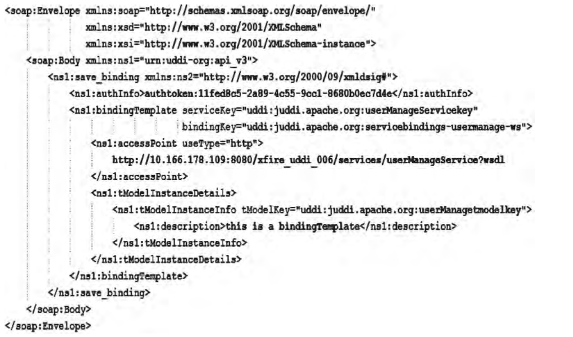


Figure 5 Process of publishing services

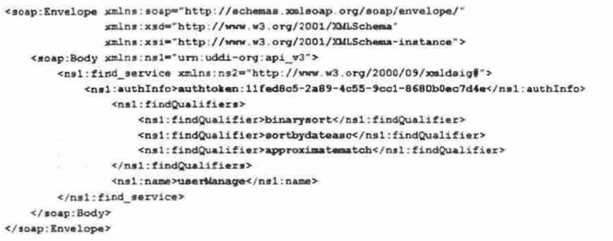
The whole publishing process needs to publish multiple entities, such as businessEntity、businessService、tModel、bindingTemplate and so on. Users can access the UDDI registry by a browser and input the specify data formatting. Registry client will convert the data to the SOAP message and send to registry server. Then, The Server will save the data into UDDI database by calling the corresponding API. Finally, the server will feed back the update results to users. Publishing data structures of UDDI are very similar, here to publish a bindingTemplate as an example. Details of the SOAP request message is shown in Figure 6.



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body xmlns:ns1="urn:uddi-org:api_v3">
        <ns1:save_binding xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
            <ns1:authInfo>authtoken:11fed8c5-2a89-4c55-9cc1-8680b0ec7d4e</ns1:authInfo>
            <ns1:bindingTemplate serviceKey="uddi:juddi.apache.org:userManageServicekey"
                                 bindingKey="uddi:juddi.apache.org:servicebindings-usermanage-ws">
                <ns1:accessPoint useType="http">
                    http://10.166.178.109:8080/xfire_uddi_006/services/userManageService?wsdl
                </ns1:accessPoint>
                <ns1:tModelInstanceDetails>
                    <ns1:tModelInstanceInfo tModelKey="uddi:juddi.apache.org:userManagetmodelkey">
                        <ns1:description>this is a bindingTemplate</ns1:description>
                    </ns1:tModelInstanceInfo>
                </ns1:tModelInstanceDetails>
            </ns1:bindingTemplate>
        </ns1:save_binding>
    </soap:Body>
</soap:Envelope>
```

Figure 6 SOAP request message of publishing a bindingTemplate

The whole inquiring process needs to find multiple entities, such as businessEntity、businessService、tModel、bindingTemplate and so on. These inquiring processes are very similar, here to find a businessService as an example. Details of the SOAP request message is shown in Figure 7.



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body xmlns:ns1="urn:uddi-org:api_v3">
        <ns1:find_service xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
            <ns1:authInfo>authtoken:11fed8c5-2a89-4c55-9cc1-8680b0ec7d4e</ns1:authInfo>
                <ns1:findQualifiers>
                    <ns1:findQualifier>binarysort</ns1:findQualifier>
                    <ns1:findQualifier>sortbydateasc</ns1:findQualifier>
                    <ns1:findQualifier>approximatematch</ns1:findQualifier>
                </ns1:findQualifiers>
                <ns1:name>userManage</ns1:name>
        </ns1:find_service>
    </soap:Body>
</soap:Envelope>
```

Figure 7　SOAP request message of inquring a BusinessService by name

4　CONCLUSION

UDDI is the key of publishing and discovering services in SOA. How to quickly find and locate the services they need is the goal of UDDI. On the basis of JUDDI, design and implement a UDDI registry based on UDDI V3 which has only one node. Moreover, verify it´s effective and achievable. In the de sign process , without considering the safety of the system and synchronizing the data in multi UDDI Registry, which are the deficiencies of the system and are the direction for future efforts.

REFERENCES:

[1] QIAO Zhiqiang, WANG Yong, ZHANG Jing. Implemention of registration centre of services based on UDDI specification [J]. Computer Enginneering and Design, 2008, 29(24): 6293 - 62955. (in Chinese).

[2] OASIS-OPEN. UDDI 3. 0 [S]. http://www. oasis-open. org/news/oasis-news_02_03_05. php. March 2005.

[3] RUI Xiongjian, WANG Zhongmin. Theory and implementation of UDDI [J]. Computer Engineering and Design, 2005, 26(6): 1602 - 1604. (in Chinese).

[4] YAO Youwen, ZHAO Lei, YANG Jiwen. The Implement Mechanism on Managing Data service Base on UDDI [J]. Control and Automation Publication Group, 2009, 25(12 - 3): 122 - 124. (in chinese).

[5] MA Xiaoxuan, HUAI Jinpeng, WANG Zhihu. Design and implementation on UDDI-based application service center [J]. Journal of Beijing University of Aeronautics and Astronautics, 2005, 31(9) 1041 - 1044 (in chinese).

[6] ZHOU Zhichao, GU Jianhua, ZHAO Tianhai, et al. Multi-domain Web Services Registry´s Research and Design Based on UDDI [J]. Science Technology and Engineering, 2010, 10(13) 3125 - 3128(in chinese).

（责任编辑：马金发）

（上接第 59 页）

[5] Bi C X, Chen J, Chen X Z. Reconstruction and prediction of multi-source acoustic field in thedistributed source boundary point method based nearfield acoustic holography [J]. SCIENCE INCHINA (Series E), 2004, 47(2): 136 - 148.

[6] Sarkissian. A Method of superposition applied to patch near-field acoustic holography [J]. Acoust. Soc. Am., 2005, 118(2): 671 - 678.

[7] 何伟，向宇，李晓妮，等. 基于多球域波叠加法的 Patch 近场声全息 [J]. 广西工学院学报，2013，24(1): 33 - 38.

[8] 王玉江，向宇，李晓妮. 多球形虚拟源强 PNAH 的声场重建精度的影响因素分析 [J]. 广西工学院学报，2013，24(4): 21 - 26.

[9] 毕传兴，陈心昭，徐亮，等. 基于等效源法的 Patch 近场声全息 [J]. 中国科学 E 辑：技术科，2007，37(9): 1205 - 1213.

[10] 毕传兴，陈心昭，周蓉，等. 球面波源叠加法与球面波源边界点法实现声全息的比较研究 [J]. 科学通报，2005，50(6): 512 - 524.

[11] [美] 罗纳德·巴罗. 企业物流管理一供应链的规划、组织与控制 [M]. 王晓东等译. 北京：机械工业出版社，2002.

（责任编辑：马金发）