REALIZATION OF WORKFLOW SERVICE INVOCATION INTERFACE FOR INTEGRATION OF AGRICULTURAL NETWORK RESOURCES

Dong Wang [1] , Ruizhi Sun [1,*]

[1] *College of Information and Electrical Engineering, China Agricultural University,No.17, Qinghua Dong Lu, Haidian, Beijing 100083, China*

[*] *Corresponding author, Address: College of Information and Electrical Engineering, China Agricultural University, No. 17, Qinghua Dong Lu, Haidian, Beijing 100083, China,Tel: 01062737945 , Email:sunrz_cn@sina.com*

Abstract: Service Invocation Interface is the interface which workflow engine uses to communicate with external application procedures. Analyzed the difficulties caused by diversity of external network resources to realize service invocation interface, took the description, search and evaluation of these resources into consideration, put forward an integrated service invocation strategy based on SOA structure. This strategy combined the invocation method for resources that could and couldn't be packaged into web services, offered the system a uniform interface to invocate various external agricultural information resources dynamically.

Keywords: Service Invocation Interface, workflow engine, Web Service, SOA Structure, resource evaluation

1. INTRODUCTION

Workflow technology separates application logic from business process logic, which gives it the ability to change system function by modifying only the business process model without involving the realization of specific module, thus makes the collaboration, integration and reorganization of the actual business to be realized swiftly. Using workflow technology in network information service area, integrate info-service resources of a

specific industry according to its business process, so that we can accomplish the collaboration and reorganization of info-service resources without changing the specific realization. Take current agricultural information service for example, the build-up of agricultural network information system now is lagged behind those developed countries to some extent, information services provided by agricultural network platforms are not in good cooperation, the problem of so called "information island" is not accidental. Therefore, to provide an integrative platform for agricultural network resources becomes a necessity, and in order to achieve this goal, we need to integrate recourses by building up a collaborative platform for heterogeneous network resources via workflow technology.

In the Workflow Reference Model raised by WFMC, a Workflow Management System (WFMS) has five unified interfaces to exchange data with other resources. These interfaces can also be called the Workflow Application Programming Interface (WAPI), among which the Service Invocation Interface identifies the invocation mechanism between WFMS and external application programs. There is a little specialty of Service Invocation Interface from other interfaces, that is it's not a necessary function of the workflow system, but once a workflow system was ineffective to support Service Invocation Interface, it would be lack of usage in practice to some extent (Chen Shi et al., 2005).

Based on SOA structure, this passage faced the integration of agricultural resources, designed an agricultural information resources collaborative platform, and would introduce in detail the design and realization of its Service Invocation Interface.

2. KEY PROBLEMS OF THE REALIZATION OF SERVICE INVOCATION INTERFACE

Workflow technology could integrate agricultural information resources of different network resource platforms, thus enabled users to check various agricultural information, gain information services of every step in the business process without knowing where exactly each of the resources came from. Therefore, how to make full use of the resources provided by these agricultural info-service platforms effectively, which was also the issue of how to design the system's service invocation interface, became the key point of the design and realization of the system.

It is not an easy job to design a uniform invocation interface for all the service resources of different kinds from various external information service platforms, and the reasons go like follows:

(1) The system needs to collaborate and integrate various agricultural information resources provided by different platforms, which use different

corresponding protocol, and have different access mechanism. So the service invocate interface has to have the ability to correspond with external resources of different kinds.

(2) The critical function of workflow management system is the routing of activities as well as the distribution of tasks. But for the real business function which should be accomplished by each task, the system could not make a prediction, only when it came to the point that the very activity got motivated, the system could decide whether external services should be called and which service was the right choice. This unpredictability of business logic results in the unpredictability of service resource interface. So that the system has to have a resource identification mechanism to identify the interface of the resource we want to invocate.

(3) Resources that need to be integrated come from various agricultural information resource platforms, the differences of these platforms in aspects of programming language and execution environment request the invocation service interface the ability to invocate cross-platform and language-independent network services.

3. REALIZATION STRATEGY OF THE SERVICE INVOCATION INTERFACE

SOA (Service-Oriented Architecture) is a component model which can connect different functional units of applied procedures (also called services) through well defined interfaces and constraints between these units. Interfaces are identified in a neutral way, apart from the hardware platforms, the operating systems and the programming languages used to realize the services. This makes the services from different platforms can interactive in a uniform and general way. It's the best choice to solve problems such as information island and legacy procedures(Liang Xiaodong et al., 2007). he realization of service invocation interface soa is only a design principle. Its structure is shown in Fig.1, which is composed of service provider, service petitioner and service register center.

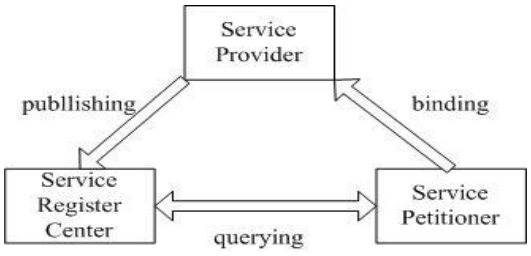

Fig.1: SOA structure

The service invocation interface web service is the best way now to support SOA structure, through its real language, platform independency and unified interface definition. And resources of different languages from different platforms can be invocated uniformly when being packaged into web services. However, there still remain quite a lot of legacy agricultural resource platforms in our country, which are inconvenient or even impossible to be packaged into web services. So, resources from these legacy resource platforms need one other strategy to be described and invocated by the way of showing their unified interfaces to service callers as independent services.

3.1 Realization of web services' invocation

Web service has inherent advantages in solving the integration of distributed applications(Ding Zhaoqing et al., 2007). The basic protocol stack of Web service is composed of SOAP, WSDL and UDDI. Simple Object Access Protocol (SOAP) provides a standard packaging structure to bind XML document to transfer protocol. Web Service Description Language (WSDL) provides a standard method to describe service interface. Uniform Description, Discovery and integration (UDDI) defines the method to discovery and publish.

The existence of SOAP makes the relation of service provider and service petitioner extremely simple: XML document exchange. For the resources which have been packaged into web services, service invocation interface only has to package related data into SOAP messages according to their WSDL documents, and then send the messages to corresponding addresses to invocate them. The structure of web service invocation interface is shown in Fig.2:

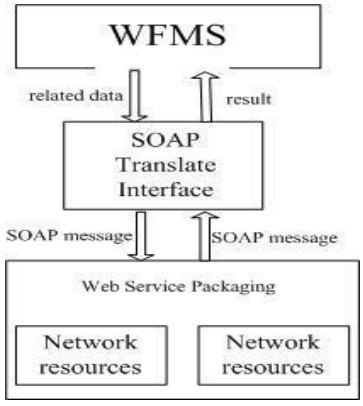

Fig.2: the structure of web service invocation interface

3.2 Realization of legacy network resources' invocation

As for legacy network information resources which cannot be packaged into web services, we divided them into independent and self-contained function units, which would be the basic units of resources integration. Although these resources can't be invocated by using SOAP message exchange, each of them would choose a way to communicate with services petitioners. And to service petitioners this difference of communication method will reflect in difference of transfer protocols.

We divided these resources into different types according to the transfer protocols they have chosen, including HTTP, FTP, RTSP etc. Although resources of the same type may be different in programming languages and platforms, they use the same transfer protocol, which means they have the same way to be corresponded with. So we would develop service invocation template for each kind of the resources. The system could accomplish the invocation of resources by using the instances of these templates.

Besides, the uncertainty of parameters when invocated these services make it necessary for the system to maintain service descriptions of the resources, including resource type, function description, uniform resource location, name of each parameter and its data type. The legacy network information resource will be described in XML form as follows:

```
- <service>
    <stype>HTTP</stype>
    <url>resource address</url>
    <description>service function description</description>
  - <param>
      <name>param1</name>
      <ptype>String</ptype>
    </param>
  - <param>
      <name>param2</name>
      <ptype>Int</ptype>
    </param>
    <rtype>String</rtype>
  </service>
```

Legacy resource invocation interface was composed by service analysis module and communication module. Fig.3 is the structure of legacy network resource invocation interface based on SOA, the concrete steps to invocate a legacy info-service resource is:

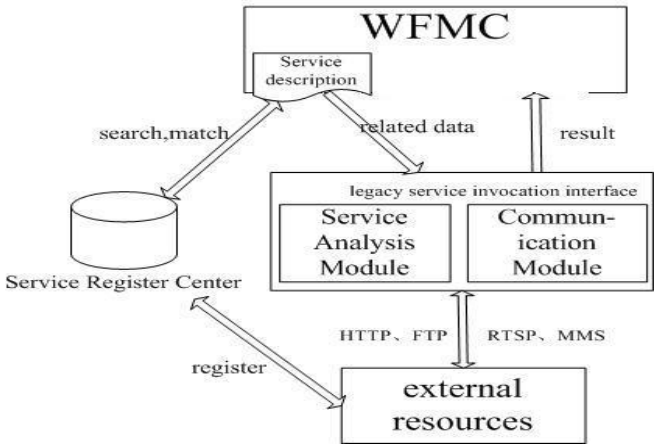

Fig.3: the structure of legacy network resource invocation interface

(1)Service analysis module analyzed the service description document sent by workflow execution logic, identified the resource type, access address, and the information of parameters etc.

(2)Communication module packaged related data sent by workflow execution logic into message of specific format, sent it to corresponding resource according to its address, and then got results sent back.

After the steps above, resources of legacy network information platforms would be seemed by workflow business process logic as independent services which approximately met the features of SOA services.

3.3 Dynamic maintenance, discovery and invocation of network resources

In order to maintain the description of legacy network resources uniformly, and further raise the reliability of web services, the system conserves a private service register center, which maintains the description of legacy network services and web services that might be invocated. Register information of each service includes the sign of service's category, the description of its function, information of each parameters and its address. And to web services, we obey the International Standard to describe them by WSDL. Compared with the web services registered in UDDI center, services registered in private service register center are more reliable, with less information redundancy, and easier to be obtained.

Services which can be invocated by the system must be the services that have been registered in UDDI or private service register center. When needed to invocate a service, the system would firstly consult the private service register center according to the service function, if a related service description was fond, then visited the resource according to the service's

category and other related data via corresponding communication method; otherwise, searched the UDDI center, looked for corresponding web service and invocated it. Besides, to the same service function, when more than one related resources were discovered, we would need a resource evaluation strategy to select the best one, we would use the strategy of alternative priority based on service category and resource responding time to decide the priority of each resource. Concretely, the method of evaluation goes like this:

(1)Every service would get a primary priority when it got registered in the private service register center. This priority was different according to different service categories. The primary priority of web service was higher than that of other network information services because of its advantages in reliability, regular and analytic of the information structure.

(2)When a service was invocated, the system would find all the resources that have similar function description with it and invocated the one had the highest priority. If failed, then the second highest would be invocated.

(3)When a resource failed in invocation, its priority would be reduced relatively.

(4)When the invocation finished, the system would remand the priority of the invocated resource according to its responding time "t". The revised value would be calculated by the formula "w=wb+K（Tr-t）",in which "wb" was the priority of the resource before invocation, "Tr" was the reference resource responding time the system set originally, amended parameter "K" was a constant used to modify the weight of resource responding time. When one resource's responding time was less than the reference responding time, its revised priority would increase after being invocated successfully, otherwise it would fall.

(5)The system had set the lowest priority values of web service resources and legacy network resources, when one resource's priority became lower than the lowest priority of its category's, it would be abandoned by the system and thus got deleted from the private service register center.

When a service registered in UDDI center was invocated successfully during the execution of a process ,which means it hadn't been registered in the private service register center, the system would add it to the private service register center automatically and gave it an initial priority; when the priority of one resource was lower than the lowest priority of its type, the system would delete it from private service register center, through this way to maintained dynamically the service descriptions in private service register center. Besides, the system maintenance staffs have privilege to add, revise and delete service descriptions in private service register center directly.

3.4 Overall realization structure of service invocation interface

Based on SOA structure, the system regarded external agricultural network resources as functional independent services. The private service register center (PSCR) maintained all the legacy network services and web services that might be in use. The system was only able to invocate the services registered in PSRC or UDDI center, and it provided a strategy to evaluate the priority of services related to the same function, thus realized the dynamic discovery, match and invocation of services. The overall structure of the realization of service invocation interface is shown in Fig.4.

We combined the two types of invocation interface above to be an integrated one when put in practice. When a service was needed, the system would check the private or the public service register center for a related service description, analyzed the description and decided which kind of service invocation type would be chosen, then sent the relative parameters packaged in certain form to corresponding address to invocate the service, and finally analyzed the result sent back from the resource and sent it back to the system. In this way, we mapped unpredictable agricultural information resources into some certain types, and integrated the invocation methods of these types of resources into one uniform interface, so that we could shield the heterogeneous of external agricultural resources for the system's business process logic.

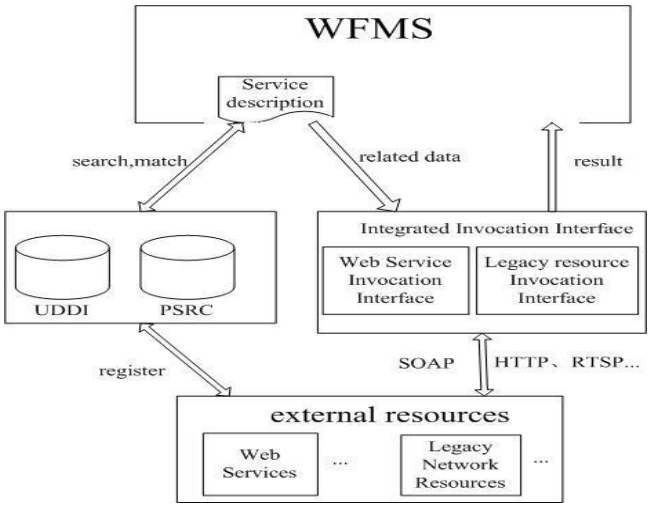

Fig.4: Structure of the overall realization of Service Invocation Interface

After isolated by the service invocation interface, all the external agricultural network resources became language and platform independent

services. And these services could be invocated through an unified way, so that the workflow business logic could effectively realize the collaboration of external agricultural network resources of different platforms.

4. CONCLUSION

Service invocation interface is the part of workflow management system which is closest to practical use. The raise of SOA structure provides a basic framework for the design and realization of workflow system's cross-platform service invocation interface. Based on SOA structure, this passage analyzed the external network info-resources' similarities and differences, raised a strategy of design and realization of the service invocation interface, and realized the collaboration of part of agricultural resources.

ACKNOWLEDGEMENTS

The research presented in this paper was partly carried out in the Institute of Geographic Sciences and Natural Resources Research, CAS and Supported by National Key Technology R&D Program No.2006BAD10A05 and China National Advanced Science & Technology (863) Plan under Contract No.2006AA10Z237. The authors are grateful to the experts of Institute of Geographic Sciences and Natural Resources Research, CAS for useful comments during the review process.

REFERENCES

Aiello M. and Dustdar S. Service Oriented Computing: Service Foundations. Proc. of the Dagstuhl Seminar 2006. Service Oriented Computing, vol. 05462. Germany, July 2006

Chen Shi, OuYang Song. Research on implement of interface 3 in workflow reference mode. Computer Engineering and Design, 2005,26(7):1862-1864(in Chinese)

Cherbakov, L., Galambos, G., Harishankar, R., Kalyana, S. andRackham, G. Impact of service orientation at the businesslevel. IBM System Journal 2005 44, 653–668

Ding Zhaoqing, Dong Chuanliang. Research on Distributed Application Integration Based on SOA. Computer Engineering, 2007,33(10):246-248(in Chinese)

Edvardson, B. and Olsson, J. Key concepts for new service development. The Service Industries Journal 1996 16(2): 140–164

Erl, T. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Upper Saddle River, NJ 2005

Ling Xiaodong. A review of SOA. Computer Applications and Software 2007,24(10):122-124(in Chinese)

Shi Meilin, Yang Guangxin, Xiang Yong, Wu Shangguang. WFMS:Workflow Management System. Chinese Journal of Computers, 1999 (3) : 326 - 328.

Workflow Management Coalition (WFMC). Workflow Process Definition Interface —XML Process Definition Language. No. WFMC - TC - 1025 , Version1. 0 , 2002

Workflow Management Coalition. Workflow Management Coalition The Workflow Reference Model. WFMC T000 -1003 ,1995

Zheng Xiao, Wan nong, Lin Guoxiang, et al. Research on microstructure of cold pressed cakes from decorticated rapeseed based on porosity, China Oils and Fats, 2004, 29(12):14-17(in Chinese)