http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (20): 5673-5676, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.5673.5676

Method for Agriculture Data Integration and Sharing Based on SOA

[1]Si Hai-Ping, [1]Shi Lei, [1]Qiao Hong-Bo, [2]Qian Shao-Jun and [3]Fang Wei
[1]College of Information and Management Science,
Henan Agricultural University, Zhengzhou, Henan, 450002, China
[2]Affairs Management Bureau of Henan Provincial Government, China
[3]Institute of Crop Science, Chinese Academy of Agricultural Sciences, Beijing, China

Abstract: For any large scale data integration application system or platform, the architecture of it decides the cooperation among different components of the system and thus decides the stability, availability and extensive ability of the system. To optimize system architecture, all factors in both internal and external environment need to be considered. In this study, with the analysis of the traditional data integration solutions and technical features of enterprise service bus (ESB) SOA_based architecture and implement approach are proposed which provides the theory basis and solid technical support for agriculture data integration and sharing.

Key words: SOA, agriculture data, data integration, ESB

INTRODUCTION

The architecture of a system refers to the components of the system and the relationships between these components. For any large scale application system, the architecture of it decides the cooperation among different components of the system and thus decides the stability, availability and extensive ability of the system. To optimize system architecture, all factors in both internal and external environment need to be considered. The purpose of this study is to study how to optimize the architecture of the current running sharing platform of agricultural science data system so that the platform system can provide more strong supports to the agricultural scientists and other agricultural science data users.

With the guidelines of the System theory, the study analyzes the SOA architecture firstly; it also analyzes the characters of the agricultural data, the users of those data and the requirements of those users. Based on the results of the analysis the author discusses the ESB data integration model and functional model and physical model of an optimized architecture, it also gives out the suggestions to the steps of improving the current architecture. Some other technologies and methodologies have been integrated into the models, such as Grid, Virtual Reality, Ant Colony Optimization, 3G technology, etc. (Abugabah and Sanzogni, 2010).

The user requirements, the technologies are keeping changing, thus the architecture of the system also needs to keep improving to catch the pace of the requirements changes. The models and suggestion in this study can provide a reference to the builders and the managers of the agriculture integration.

WEB SERVICE AND SOA ARCHITECTURE

SOA is an architectural approach to creating systems built from autonomous services (Bravo *et al*., 2006). With SOA, integration becomes forethought rather than afterthought-the end solution is likely to be composed of services developed in different programming languages, hosted on disparate platforms with a variety of security models and business processes. While this concept sounds incredibly complex it is not new-some may argue that SOA evolved out of the experiences associated with designing and developing distributed systems based on previously available technologies. Many of the concepts associated with SOA such as services, discovery and late binding were associated with CORBA and DCOM. Similarly, many service design principles have much in common with earlier OOA/OOD techniques based upon encapsulation, abstraction and clearly defined interfaces (Chien and Barthorpe, 2010).

SOA, based on Web Services, promises to simplify integration by providing universal connectivity to existing systems and data. SOA has already been widely recognized as an effective paradigm in order to achieving integration of diverse information systems.

SOA-based architecture can cross boundaries of platforms, operation systems and proprietary data standards, commonly through the usage of Web services

Corresponding Author: Fang Wei, Institute of Crop Science, Chinese Academy of Agricultural Sciences, Beijing, 100081, China

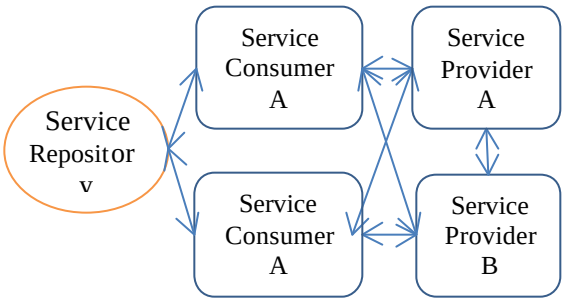

Fig. 1: Typical SOA model

technology. Thus, the design in our system used SOA in that the features of flexibility and cross-platform. Typical Service oriented architecture includes service provider, service registry, service consumer three kinds of roles. Services need to be registered in the registration center in accordance with service contracts to register. Service contracts include contracts, authorization, conditions that can be used and the information of service providers. Process or information was encapsulated by service provider. With a platform and language, independent the internal and external interfaces were defined. And then the interface description information released to the service registry that service requester can be used.

PATTERNS IMPLEMENTATION FOR AGRICULTURE INTEGRATION

There is several design patterns aimed at support of the enterprise data in SOA implementations, some of which are well established and some are emerging, based on the experiences in the current SOA implementations. One of the most popular approaches to SOA implementation today is usage of the enterprise service bus pattern, allowing to "virtualizing" enterprise services access b. Similarly, enterprise data bus allows for virtualization of the enterprise data access.

ESB is used as a service container implementation approach and XML service interface is used to access various heterogeneous data sources. (Dewiyanti *et al*., 2007). Extensible style sheet Language transformations (XSLT) is used for data mapping between heterogeneous data (Hwang *et al*., 2006). ESB provides functions of heterogeneous data extraction, data conversion, data exchange standards, patterns and exchange formats. By calling the different services, the corresponding functional block are executed (Stefansson, 2004). When external data need to be integrated, the first step is to connect to the database system and select different elements needed from the data schema and bind them to

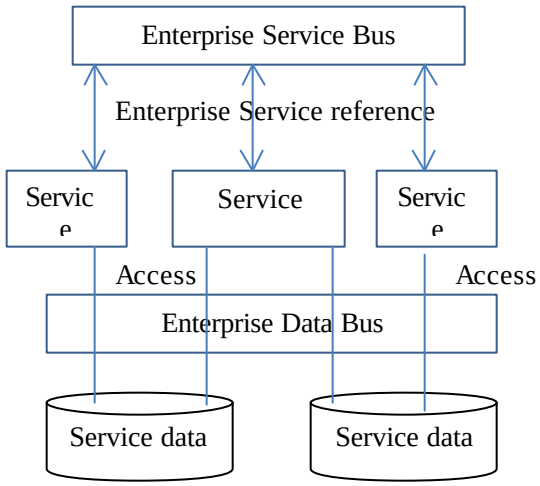

Fig. 2: Enterprise service bus data integration model

component fields within an application. The different elements that comprise the data architecture are identified. Data binding is represented by the arrows between the components. Data bindings need to be set up between properties of UI controls and properties of a data set component and between a resolver and a connector component. (Triantafillou *et al*., 2003)

In a lot of ways this design pattern is similar to the enterprise data access as a kind of service pattern. The difference is that instead of promoting data access to a first class business service, in this case it is treated on the level of the integration layer, thus providing a direct access from any service to any piece of the enterprise data Fig. 2.

The implementation adds additional layer-enterprise data bus, providing access to the enterprise data, contained in either enterprise applications or their subordinated databases. It also modifies implementation of the business components layer; in this implementation it does not contain "wrapper" components any more. Business components, in this case are comprised only of the components implementing service functionality based on the existing functionality and data accessible through the enterprise data bus. The advantages of this pattern are:

Explicit separation of concerns between implementation of the service functionality (business logic) and enterprise data support logic (similar to the previous pattern). Enterprise data bus effectively create abstraction layer, shielding business functionality from details enterprise data/functionality access.

Through encapsulating all of the access to enterprise data/functionality enterprise data bus provides a single

place for all of the transformations between enterprise semantic data model and data models of enterprise applications.

Because any of the service implementation, in this case, has an access to any of the enterprise data it requires, such implementation allows to significantly reduce coupling between service-service invocation contains only data references (key) which change extremely rarely while the actual data access is implemented by the service itself using enterprise data bus. This means that if the service implementation requires additional data for its processing, it can access it directly without impacting its consumers.

Due to the amount of accesses (synchronous) performance becomes one the most important characteristics of the enterprise data bus. Any performance degradation of the bus can effectively destroy SOA implementation.

DESIGN AND IMPLEMENTATION FOR AGRICULTURE INTEGRATION BASED ON SOA ARCHITECTURE

The architecture design of the integration for agriculture data resource is based on SOA and the service realization through Web Services. The functions of each module are defined as independent services. Integration business processes complete by the work of a single service or a combination of multiple services. So, the presentation layer provides a holistic view of business logic. Improvement of the reusability is the system an important goal of architecture design. The varieties of services are given in the form of application programming interfaces (API) and integrate in a loosely coupled way to achieve the purpose of reuse. The application programming interface describes by WSDL and uses SOAP transmission as a bridge to interact with the outside world.

In the design, the function implement in agriculture data integration are centered around definition of services. With service-oriented decomposition, based on the business processes, functional model, long term architectural goals and reuse of the existing functionality are required. This approach usually incorporates one of the most important assets of the agriculture data usage as an afterthought. In this article, SOA_based architecture is introduced; design patterns for incorporating of this data and the complexities of dealing with the agriculture data are outlined. The Overall system architecture and service deployment are shown in Fig. 3.

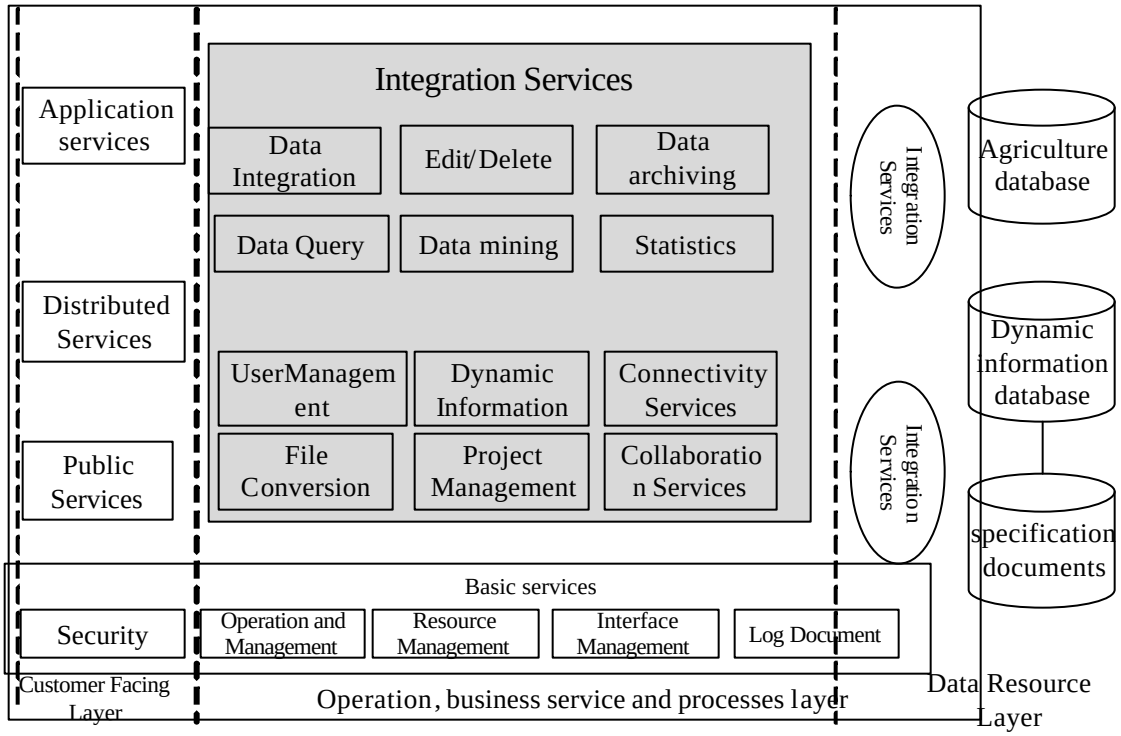

Fig. 3: SOA_based architecture for agriculture data integration

Such design defines, as shown in Fig. 2, multiple layers in the SOA architecture leads to implementation of the data integration services in a form of a specialized layer rationalizing existing functionality (applications) against "ideal" enterprise business model.

Data resources and operational layer: It represents the portfolio of existing applications (i.e., legacy, COTS and custom built systems).

Integration layer: This layer uses various technologies to expose existing enterprise resources and operational systems so they could be used by business components.

Business components layer: Function Business components are deployable units of software that provide the functionality required by the business services. These components can be either newly developed or "wrappers" using integration layer to access the functionality of existing enterprise resources.

Business services, processes layer: Integration services provide high level business functionality throughout the enterprise. This layer effectively bridges between the "ideal" business model and the existing enterprise IT assets-applications and business components. Integration Business processes allow for creation of business solutions through orchestration of business services.

Customer facing layer: User facing provide support for users (both inside and outside the enterprise) to view and control the execution of the enterprise business processes and/or services. These users can be either humans using Web or rich clients or B2B connections supporting intra enterprise business processes.

AGRICULTURE DATA ACCESS ISSUES

Although the SOA_base architecture implementations hide enterprise data behind service interfaces, data access issues are still required solving:

Consolidation of data between multiple applications: Today's agriculture data is typically scattered between multiple applications. In addition, data representations themselves are different between different applications. As a result, it is often difficult to reconcile data representation between individual applications. In the SOA implementation, a well-defined enterprise data model in introduced to represent enterprise-wide functionality it needs to operate.

Definition of interface: Since service invocations are always remote, the service design takes kinds of large granularity interfaces, in order to minimize the amount of service traffic (chattiness) between service consumers and providers.

CONCLUSION

As the scope of SOA implementations expands from limited in scope departmental implementations to widely use undertaking the issues of data access are quickly starting to become one of the most important implementation issues. If there is not architected correctly from the very beginning, agriculture big data access can become a major problem down the road.

ACKNOWLEGMENTS

The authors would like to thank for the support The '12th Five-Year Plan' in the Rural Areas of National Science and Technology Plan Project in China (2013BAD15B0602, 2014BAD10B06), the author also thank for sport by Henan Province Special Major Science and Technology Project (121100110900) and the project of Department of Agriculture Information Service Technology Key Laboratory (2013-AIST-03).

REFERENCES

Abugabah, A.J.A. and L. Sanzogni, 2010. Re-conceptualizing information systems models: An experience from ERP systems environment. Int. J. Infonomics, 3: 414-421.

Bravo, C., M.A. Redondo, M. Manuel Ortega and M. Felisa Verdejo, 2006. Collaborative environments for the learning of design: A model and a case study in domotics. Comput. Educ., 46: 152-173.

Chien, H.J. and S. Barthorpe, 2010. The current state of information and communication technology usage by small and medium Taiwanese construction companies. J. Inf. Technol, Constr., 15: 75-85.

Dewiyanti, S., S. Brand-Gruwel, W. Jochems and N.J. Broers, 2007. Students' experiences with collaborative learning in asynchronous computer-supported collaborative learning environments. Comput. Human Behav., 23: 496-514.

Hwang, W.Y., N.S. Chen and R.L. Hsu, 2006. Development and evaluation of multimedia whiteboard system for improving mathematical problem solving. Comput. Educ., 46: 105-121.

Stefansson, G., 2004. The tutor-web: An educational system for classroom presentation, evaluation and self-study. Comput. Educ., 43: 315-343.

Triantafillou, E., A. Pomportis and S. Demetriadis, 2003. The design and the formative evaluation of an adaptive educational system based on cognitive styles. Compt. Educ., 41: 87-103.