Journal of Theoretical and Applied Information Technology
31st December 2012. Vol. 46 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A DEVELOPMENT MODEL FOR DOMAIN-ORIENTED INFORMATION SERVICE BASED ON CLOUD COMPUTING INFRASTRUCTURE

[1,2]ZHENG HUA

[1]School of Management and Engineering, Nanjing University, China

[2]Assoc Prof., School of Information and Statistics, Guangxi University of Finance and Economics, China

E-mail: gxhuazheng@yahoo.com.cn

ABSTRACT

For the diversity of users demand for information services in converged network environment, a solution was proposed with the combination of cloud computing technology and domain engineering, in which three stages starting from the domain engineering include service-oriented domain analysis and modeling, domain architecture design based on cloud computing infrastructure and construction of service-oriented domain infrastructure. It will achieve the development, assembly and reuse of large-scale distributed information services in the domain by fully using cloud computing technology in the advantages of integration in the business.

Keywords: *Converged Network, Cloud Computing, Domain Engineering, Information Service, Component*

1. INTRODUCTION

Jean T. Sutcliffe points out in his book 'Measuring Information: An Information Services Perspective' that information services are a kind of resource formed by an independent institution or a specified function of the institution, and its purpose is to provide information for the user groups [1]. In recent years, with the emergence of Web2.0 and mobile network technology, the internet is showing the web-based features of network. Nowadays, information services are faced with a more complex, heterogeneous, open and changing converged network environment. The deployment and run of each information service all must rely on some environmental factors such as a certain network, resources and so on. Meanwhile, a large diversity uncertainty of user requirements has also led to frequent change and multi-target user needs. The future direction of development of information services is to co-construct value-added services to meet user needs by using a number of loosely coupled services in a variety of cross-platform network environment. In services design for the new user needs, on the one hand it is ensured that services designed can be easily reused by others; the other ensured that existing services can be reused. This relates to how to build a services development model of meeting user requirements and easily reusing other services. Although the new service models and technologies are emerging such as SOA, virtualization, distributed computing, grid, cloud computing and so on, because of the lack of an effective methodology, the further use and development of information services are still restricted.

2. RELATED WORKS

Chen Jianlong thoroughly analyzes the various elements of the information services and its basic model, generated models etc. He specially emphasizes that information services activities are based on the user-oriented[2]. Liu Kunxiong pointed out that the needs of user information are showing some characteristics in the current network environment such as community, diversity, and dynamic nature of real-time, quality and efficiency and so on[3]. Xia Nanqiang studied some issues related to information services in converged network environment [4]. Liu Gaoyong pointed out that in Web2.0 environment information services are user-centric, and made four kinds of expansion modes. From the existing achievements, scholars are more focused on related elements and internal mechanism of information services from the perspective of pattern, and stressed the importance of user needs, but the study of information services development model in converged network environment is still less. How to develop

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

information services to meet the diversity needs of users in converged network environment is still a problem to be solved.

Cloud Computing [6] first proposed by Google is a new resource model. Its main feature is that the system owns large data sets, and services are supplied to the users based on the data set. Cloud computing uses a lot of ordinary commercial machine to build the system, simultaneously reliability and availability of overall system are ensured by the redundant storage [7]. Domain is functional areas covered by a group of application systems with similar require [8]. Domain engineering emphasized that all activities in field application are completed by the combination of domain knowledge and engineering. In converged network environment, the development of information services can take advantage of domain engineering ideas. The basic idea is: according to three stages of domain engineering, domain analysis, domain design and domain implementation are used to identify, develop and assemble reusable components. When a information service system is constructed, first, requirements specification for new applications is determined according to the domain model, then the design of new applications is formatted based on domain-specific software architecture, finally from the component library to obtain the necessary resources, and by component assembly approach the required application is obtained.

Based on the above analysis, because of the complexity of the network environment and the diversity of user needs, these cause that all kinds of information services is difficult to develop and reuse, limit the further development of information services. To solve these problems, in this paper the ideas of domain engineering combined with cloud computing technology are introduced to study how to use its unique domain knowledge to design and implement effective information services to meet user needs for a specific application areas.

3. THE DOMAIN ORIENTED INFORMATION SERVICE DEVELOPMENT MODEL

Converged network is a more open, complex heterogeneous network environment, in which an information service to meet user needs are often deployed and implemented by spreading across networks and multiple organizations. As information services face the environment including heterogeneous, open, complex, changeable and other features, it bring a variety of uncertainties for the run of information services, and make the information services performance and quality difficult to be guaranteed. Therefore, in the converged network environment, the methods and techniques of traditional analysis, design and implementation are difficult to adapt to local self-government, self-coordination, and dynamic evolution and so on.

As the multi-user demand problem is very complex in the heterogeneous environment, information services discussed in this paper are firstly defined in the context of a particular area. Here, we regard some kind of information services as the form of the software realization for the business process by component. Compared with other systems within the domain, some application, while having its own characteristics and needs, but has more of a commonality. By drawing on the common domain knowledge, software system components are constantly revised in accordance with the principles of software reuse, and reusable components are gradually collected and developed, finally a prototype system is gradually formed. In converged network environment, there is a gap between the prototype system and the user's individual requirements. So, as the growing awareness of the developer for business processes, a prototype system also need to constantly be supplemented and improved. Taking into account the cost of software development and software reuse inadequate mechanisms, system development is also mainly for the purpose to meet current user needs. With the increase and accumulation in domain knowledge, a more mature domain architecture and application framework will be gradually formed, and more of the software section gradually evolved into the software reusable components with higher degrees. Referring to domain engineering and component-based development thinking and cloud computing technology in the advantages of business integration, this paper presents a domain-oriented information service development model. The model can be divided into service-oriented domain analysis and modeling, cloud computing based domain architecture design and service-oriented domain reuse infrastructure three stages, covering the domain analysis, domain design and domain achievement three stages. The implementation processes are shown in Figure 1.

Journal of Theoretical and Applied Information Technology
31st December 2012. Vol. 46 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

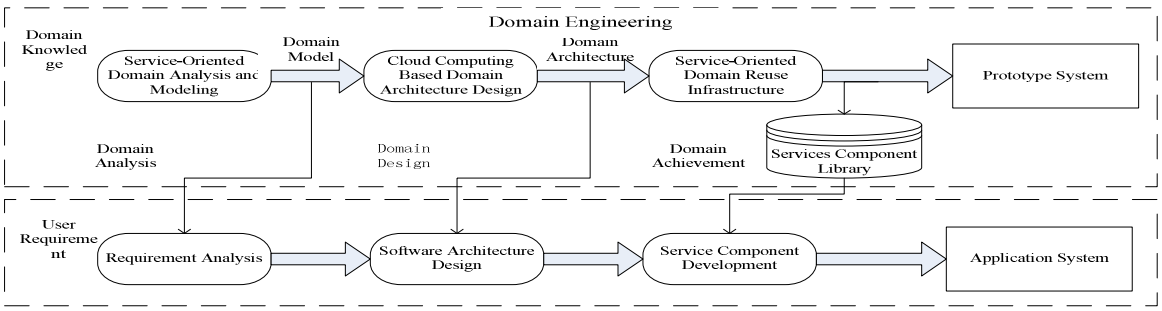

Figure.1 The Domain-Oriented Information Service Development Model

The domain prototype system for information service is ultimately developed based on the above work of three stages in a converged network environment. Typically, these three stages are sequential. But in the actual operation of the process, it is iterative. Throughout the development model, application system development focuses on a single system, while the focus of domain engineering is more than a related system in one domain. The results of the domain engineering can be applied to application system development, and application systems development activities, in turn, affect future activities of the domain engineering.

4 .MODEL IMPLEMENTATION

The development model focuses on three stages. Its essence is enabling reuse-based component development [9][10]. The implementation of the model lies in how the idea of the domain engineering applied to the development of information services. The following three stages will be described and analyzed.

4.1 Service-Oriented Domain Analysis and Modeling

Domain analysis is an activity in which similarities and differences of user needs are found and expressed through analysis and study of the characteristics of a group of applications. Its purpose is to define the boundaries of the domain, and obtain the user's demand model. In the service-oriented domain analysis and modeling stage, First, the developer obtain the domain knowledge from experts in the domain, and then use the existing domain knowledge to help to analyze the user needs and obtain reusable requirements in this domain, and ultimately the specification of user requirements is formatted, the demand model of domain is constructed.

Currently, there are many domain analysis methods owned advantages and disadvantages such as object-oriented domain analysis method (OODA), Feature-Oriented Domain Analysis method (FODA), Organizational Domain Analytical Modeling Approach (ODM) and so on. At this stage the most used method is FODA, but it lacks the sophisticated tools can be used. UML model is a more mature product, which can be used to implement service-oriented domain analysis and modeling. The main idea is to gather requirements and conduct the demand analysis of domain, then use the use case model, static model, dynamic model to establish the demand model of domain.

4.2 The Domain Architecture Design Based on Cloud Computing Architecture

The environment has significant impact on the information services, and it is the key by which information services are to achieve normal operation and optimum operation. Domain architecture is related to technology environment and software infrastructure, and also includes the software platform of the system architecture, which is the composition of the business component. As a basis solution is provided to implement reusable components for the next phase, and it is a higher level of design. In the design of domain architecture, system-level design issues are be considered. There are the typical features, such as capacity, throughput, consistency, compatibility, etc. the architect based on cloud computing supplies possibility for large-grained software reuse.

By considering the system environment (such as operating systems, databases, communication mechanisms, middleware, etc.) and following the standards and reserving space for the variability, the domain architecture designed based on cloud computing architecture is shown in Figure 2. In this structure, the service component resources are deployed in the cloud. Front-end user interface allows the user through the service directory to find and select the desired service. When the service

Journal of Theoretical and Applied Information Technology
31st December 2012. Vol. 46 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

request is validated, the right service resources are found by the system manager, and then call the service tools to tap the resources of the cloud services. In the service implementation process, the services tool need to go through the system manager for resource monitoring and configuration. The domain architecture based on cloud computing can avoid coupling system components, and achieve through the component as a service transparent to users in network, and the user just pay for related services based on actual use of the resources. This architecture can improve the development efficiency, and reduce system complexity, and increase system scalability, re-configurability and maintainability.

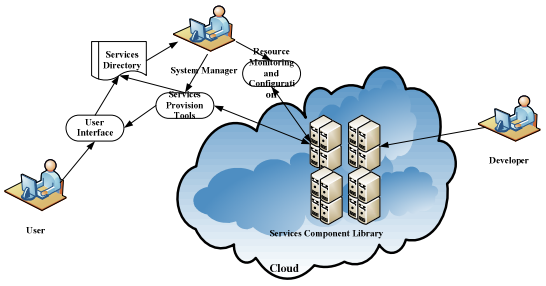

Figure.2 The Domain Architecture Based on Cloud Computing Architecture

4.3 The Service-Oriented Domain Reuse Infrastructure Building

This phase corresponds to domain implementation in the domain engineering. Its goal is for the development of reusable services components (design, coding, testing and other activities), and for efficient architecture-based reuse by storing into the domain service component library. After the previous stage to get the system component architecture, this stage is achieved by coding. At present there are three major technical standards: Microsoft's COM/DCOM; OMG's CORBA; SUN's Java Bean /EJB. The main choice of development tools for service components to achieve is JAVA and. NET. In a project of domain engineering, there is being different implementation styles for different components. Its core idea is to use existing components and components that will be developed to quickly assemble high-quality applications.

Any information service process can be seen as a collection of a series of orderly business. These operations are encapsulated in the service component, and specific business is achieved through the operation of the service components. Most component assembly is based on the workflow mechanism (such as BPEL4WS, etc.) and Petri nets. For this we extract the business process in the domain model, at the same time the features reflecting domain characteristics are included in this process, and ultimately the domain assembly model is constructed。It represents a group of the system business process of commonality and variability in this domain. Which services are required, optional, and one of many that will be determined by the domain assembly model and specific user demands. This will help guide the assembly of service components.

For large-scale service component resources deployed in the cloud, how to achieve a distributed call is an important issue. The MapReduce programming model for parallel computing is introduced in this paper. A simplest MapReduce application contains at least three parts: a Map function and a Reduce function and a main function. First, the service component will need to be sent to multiple nodes, then be executed paralleled in each node. Finally, the result will be integrated into the output. While the output of the first phase can be used as the input for implementation of the next phase, so a tree diagram of distributed computing can be imagined, which is shown in Figure 3.

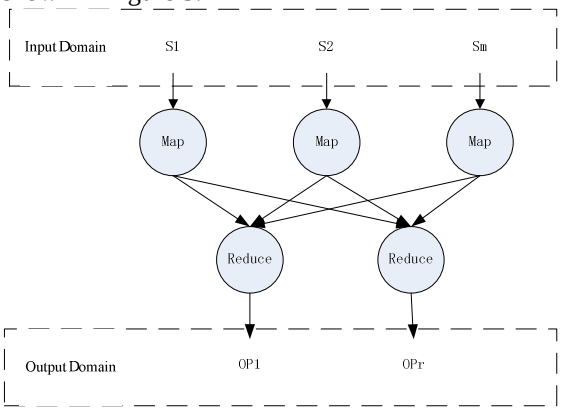

Figure.3 The Component Execution Model Based on MapReduce

5. CONCLUSIONS

This paper presents the information service development model combining key technologies including domain engineering and component-based development, and fully integrating the advantages of cloud computing technology in business integration, and divided into three levels including demands, design, and component. It will allow developers to quickly build applications and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to maintain relatively stable, and reduce the development and maintenance effort, and achieve the development, assembly and reuse of large-scale distributed information services within the domain.

The next step is: how to closely link the domain engineering to the information service system in various domains, and how the research of domain engineering can be used to improve the efficiency of system development to better meet the needs of users.

ACKNOWLEDGMENT

This work is supported by the 2011 Maintain School Safety and Stability Fund Project of Guangxi province, China(class A).

REFRENCES:

[1] J.T.Sutcliffe, *Measuring Information: An Information Services Perspective*. San Diego: Academic Press, 1995.

[2]Chen Jianlong, "A Study of Information Service Models". *Journal of Peking University(Philosophy and Social Sciences)*, Vol. 40, No. 3,2003,pp.124-132.

[3]Liu Kunxiong,WANG Xiuli, "Theoretical Study of Network-based Transformation of Information Service", *Information Studies:Theory & Application*, Vol.32, No. 10,2009,pp.40-44.

[4] Xia Nanqiang,Yin Ketao, "Thought on Information Service Research under the Network Convergence Environment", *Information Studies:Theory & Application*, Vol. 33, No. 7,2010,pp.31-34.

[5] Liu Gaoyong,Wang Huiling, "The Innovation of Information Service in Web2.0 Environment", *Library and Information Service*, Vol. 53, No. 7,2009,pp.39-42,8.

[6]Chen Kang, Zheng Weiming, "Cloud Computing: System Instances and Current Research", *Journal of Software*, Vol.20, No. 5,2009,pp.1337-1348.

[7] MiChAEL ARmBRuSt, Armando fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No. UCB/EECS-2009-28, http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

[8] Li Keqin, Chen Zhaoliang, Mei Hong et al, "An Introduction to Domain Engineering", *Computer Science*, Vol. 26, No. 5,1999,pp.21-25.

[9] Richard W. Selby,"Enabling Reuse-Based Software Development of Large-Scale Systems," *IEEE Transaction of Software Engineering*, Vol. 31, No. 6, 2005,PP. 495-510.

[10] Marcus A. Rothnberger, Kevin j. Dooleg, Uday R. Kulkarni and Nader Nada, "Strategies of Software Reuse : A Principal Component Analysis of Reuse Practices", *IEEE Trans. Software Eng.*, vol. 29, no. 9, 2003,pp. 825-837.