RAMS Architecture based on Cloud Service

Hanyong Choi, Sungho Sim

Abstract: The architecture structure for computing resource management is rapidly changing based on the cloud in the client - server environment. Also, software development is becoming design-oriented, compared to the code-oriented architecture processes. The study proposes an architectural model to provide asset service based on cloud service for change of software production method and efficiency. The proposed architecture model enables creation of design assets suitable for design domain based on reusable standardized architecture. The Reusable Asset Management System (RAMS) consists of 4 tiers of asset service architecture. In addition, RAMS-based design assets can acquire information on new services and design quality when selecting registered asset services in addition to services selected by the method of collecting dynamic information according to client request.

Index Terms*: *RAMS, Design, Cloud, Architecture, Asset.

I. INTRODUCTION

The software market is evolving from a market to support specific industrial areas to an area for designing new industrial structures. It is predicted that the changes in the software market will be diversified more than now, centering on the fourth industrial revolution. This is the time to not only consider the change in production method, but also the efficiency of production[1]. In order to solve this problem, open platforms were introduced based on open source and application of system was applied to various areas by using this platform. To this end, many methods have been researched and developed for effective use of existing systems or resources according to the generalization of services. However, interconnection in a distributed environment has various difficulties and problems. It focuses on elements of physical network environment rather than consideration of design information of designers. The study intended to reuse design information based on the cloud model to meet the demand for well - abstracted design information in system development[2]. As a precedent study to solve this problem, a component structure that can reuse architecture - based models and design information was designed[3].

In the previous study, an architecture to build a production line method based on a standardized platform as a manufacturing method of small quantity production method based on PLE was used. Based on this, small-scale software of various fields required in the fourth industrial revolution can be produced on a timely basis based on a standard model. In addition, this study intended to use the cloud-based architecture to use the domain design information as a component asset in the design stage independent of the development environment in the existing proposed model. Using an architecture based on open source, the goal is to provide an optimized environment for extending the assets to the intended use of the designer[4].

To do this, it was attempted to restructure the existing repository structure based on the cloud. The design information corresponding to each domain service area in the cloud can be structured as big data, and it is possible to formulate a model suitable for various products.

Therefore, according to the results of this study, it will be possible to provide a solution for how to provide needs of designers for various architectures in order to support cloud service within the cloud model base. In addition, it can accommodate platform - optimized assets, and it is possible to construct appropriate design information according to the designer requirements on the same platform.

II. RELATED WORKS

A. PLE Development Method

Various studies based on PLE have been conducted to improve productivity for software production[5,6]. Due to changes in the environment, academia and industry are researching and developing cloud-based software production technology, and are supporting various services based on this. In the industry, Italia Telecom, Northern Telecom, HPC, and Robert Bosch GmbH conduct area analysis and product line analysis, and research is actively being carried out in the academic community by Software Engineering Institute, Technical University of Ilmenau and POSTECH. In addition, many domain analysis methods use feature-based analysis methods to analyze commonalities and differences in domains. ODM, FeatureRSEB, FODAcom, and DEMRA are typical examples of such research. Feature Oriented Reuse Method (FORM) is a product line development methodology that constructs a feature model from requirements and then produces a product based on the feature model. The feature model is a means for expressing feature level commonality and variability and has been developed to cope with various environmental changes of software with formal analysis.

B. Cloud Computing Architecture

To understand the cloud platform, one must consider the different levels of cloud computing involved. Also, as with most technologies, one should consider advantages and disadvantages and define features that differ from other technologies[7]. Different levels of cloud

Revised Manuscript Received on December 22, 2018.

Hanyong Choi1, Shinhan University, Dongducheon, Korea
Sungho Sim, Semyung University, Jecheon, Korea

computing include various services and deployment models. Each of these models allows versatility to meet the needs of different customer bases.

On a higher level the cloud architecture can be used to describe each component of cloud computing ranging from the physical servers and networks, the middleware, and even to individual application function that are more commonly associated by clients for cloud computing.

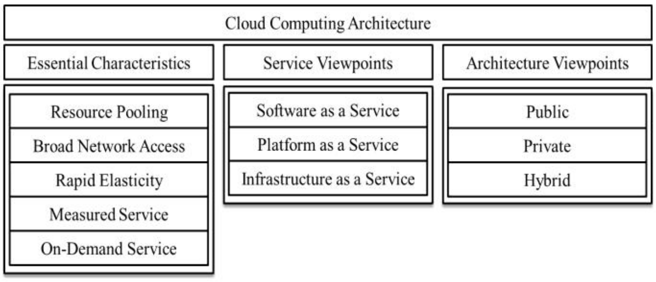

Fig 1 Cloud Computing Architecture

Figure 1 shows a general overview of each architecture layer in cloud computing. Cloud computing can be divided into three levels at each tier that represents a core part of cloud computing[8-11]. The application layer contains data and a variety of applications used by clients, such as a web interface, a programming interface for developing applications, and a native engine for cloud applications called application cores. The virtualization layer provides the necessary resources and demands to back up the application.

Database access and server functionality is found not only in this layer but also in connectivity components such as Internet Protocol and Domain Name System (DNS). In addition, virtualization of the physical infrastructure is also provided where everything related to the virtual machine is defined and controlled from the virtual component. The physical layer is the hardware and resources used to perform the required tasks in the two tiers above. Hardware includes individual servers, and facilities that accept hardware are also part of this layer and are responsible for proper maintenance of the physical components.

III. RESULTS AND DISCUSSIONS

Cloud based Asset Management results

A. Service Model

This study proposes a method to support suitable cloud service to support designer requirement for various architectures based on cloud model.

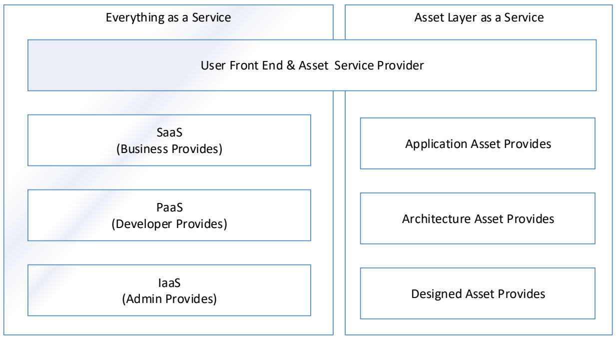

Fig 2 Basic Model for Service Asset in Cloud Architecture

Thus, it becomes possible to accommodate optimized assets based on the platform, and it is possible to construct appropriate design information according to the designer demand in the same platform. When the cloud-based model of Figure 2 applied in this study is divided into three tiers, each tier is a basic model of the cloud architecture service on how to construct the asset structure to process specific SaaS, PaaS, and IaaS.

B. Service Model Considerations

Cloud computing is a service that is provided to users through the Internet by integrating IT resources that exist in different physical locations into virtualization technology. The assets that are to be provided to designers are based on cloud services and they must be considered to include features such as virtualization, hiring, scalability, standardization, and automation and energy efficiency. The necessary considerations for providing the asset using the proposed model structure are shown in Figure 3.

- General SLA opening method for asset service of asset service server
- For asset service client, service broker transfer method according to designer requirement
- For service broker, SLA collector request method to find a selectable service
- For SLA collector, provision method of cloud service list
- Real-time evaluation method for each asset service

Fig 3 Considerations for asset provision

At this time, the SLA (Service Level Agreement) structure should describe the minimum level and the measurement standard for the request for the service to be provided with the IT service and the clear problem solving method between the service provider and the service user, and it should be designed so that promises made through prior discussions are possible[12,13]. Therefore, the SLA should have a structure that can include content that should provide the level of services and services that directly affect the designer of the application domain as a client of the operational business through the discussion process.

C. RAMS Architecture

The existing DMC architecture based on the repository is a three-layer structure for transforming

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

design information and formatting it in the repository[2]. In addition, cloud-based Reusable Asset Management System (RAMS) system architecture is designed as a four-layer structure for managing the design asset information with added cloud layer as shown in Figure 4. The Design Layer is a layer for designing asset information based on UML and is an upper layer for reusing existing design assets. Infra Layer is a layer for reconstructing and storing architectural design assets and component information in XML. The cloud layer reuses the design information of standardized assets by domain and manages access to standard and non-standard information.

The information between the asset design information and the design layer is a layer for transforming UML representation and XML information using a mapping layer, and is responsible for managing assets and generating codes. The mapping layer provides a module for modeling the design structure represented by UML and a service for storing meta models constituting the design structure of the infrastructure layer. It is composed of a code generation module that generates XML code by mapping stored meta models to a code generation template. The designer is divided into the component designer and the architecture designer for authentication, and the component designer registers the service component and is authorized to the registered service component. The registered service component supports faceted search with faceted items of classification by use purpose and evaluation besides searching existing component name. In order to create a design structure of a reusable architecture asset that is independent of specific tools and platforms, the architecture information is registered with the architectural asset designer, and the structure is modeled as UML and the configuration information is stored in the product line asset. In addition, it supports a UML-based editor that can model such a design structure visually.

In order to reuse the architecture structure in the future, the configuration information stored in the database is mapped to a code generation template composed of the XMI specification, and the generated XML code is stored. At this time, according to the attributes of each model assigned to the ray, the format of the asset to be displayed in the model of the layer is determined. In the cloud layer architecture model, public, private, and hybrid attribute values are determined. Also, in the service level, SaaS and PaaS are provided in the service model of the design stage, and IaaS is additionally provided in the infrastructure layer. However, since the boundaries of these service methods gradually disappear, in cloud-based environments, these services can be provided as integrated services.

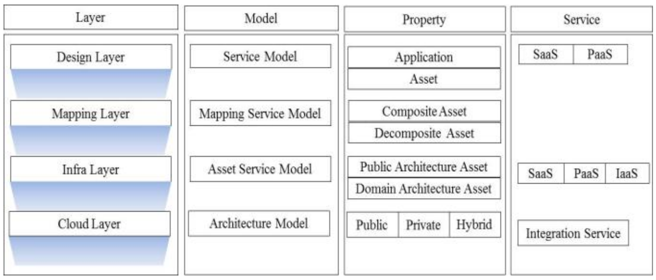

Fig 4 Cloud based RAMS Architecture

Generally, the repository used by the design tool reuses the E-R model or OO model design information as a data model. However, at this time, the stored design information has been standardized but the architecture has not been formalized. Therefore, in this study, the design information to be stored in the information repository is formulated and stored, and it can be reused on the cloud basis. In order to facilitate the reuse of design information, standardized design information was stored in the architecture area of the repository. Also, in order to reduce the redundancy of design assets in the cloud, asset design information is stored using the asset class structure. Therefore, the business context and properties are first determined according to the procedure of DMIC asset development and management method in Figure 5. It then chooses the specification of the asset and chooses whether to reconfigure or register the new asset. In the case of reconfiguration, after defining the interface, the asset is defined and tested, the new asset is implemented and packaged, and finally the quality of the asset is verified.

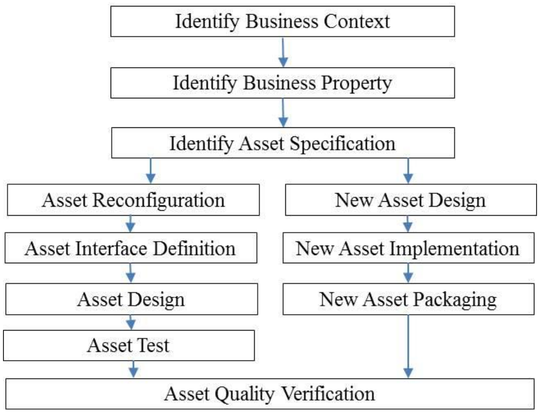

Fig 5 Asset Development and Management

At this time, the system designs and stores the architectural assets of the standardized software in the cloud in order to reuse the design information. Because the architecture assets of the cloud are designed to use a hybrid structure that can be duplicated by separating common product line asset information into architecture and service component areas, the architecture area, which is a standardized design asset area, can represent the domain information in a standardized manner and set the service method as the characteristic value. Therefore, by using the design information stored in the architecture area based on the cloud differently

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

according to the service method, the designer can design the system by configuring the product asset in a standardized manner. At this time, the design information represented by the characteristics on the user design domain may be classified into the architectural design information of the domain to be designed, and stored as an asset of the service component area, and reused. The redundant structure of the cloud-based PLE asset preserving design information can maintain the independence of the domain to be designed and the standardization of the design information, and it has flexibility according to the characteristics of the design domain.

CONCLUSION

It is difficult to share design information flexibly with existing architectures that independently use assets based on storage, and it is difficult to design assets based on securing the required quality in the fourth industrial revolution. In order to solve this problem, this study proposed a cloud model for sharing standardized design assets. Cloud computing supports low-cost services to enable efficient use of resources.

It is also possible to expand the scope and scale of services by combining services with cooperation between service providers. Asset design methods based on cloud services can provide availability, reliability, and high quality by providing only services for designers' design assets. In addition, the SLA of the cloud service for the assets constituted in this study guarantees such designer requirements. At this time, the reach of the SLA differs for each service and can be different from the expectation level of the designer.

Therefore, by using the study result, the cloud - based asset system is able to create a design model suitable for new service requirements by shifting from the provider - oriented service, which was a problem in the existing method, to the position of the designer. Further, when selecting the registered asset service in addition to the service selected by the method of collecting the dynamic information according to the client request, the information about the new service and the quality can be strengthened. As future research goals, in order to ensure the quality of assets under international standard procedures, it is determined to be necessary to study the verification method for securing the quality of standardized design asset information.

REFERENCES

1. Sudha Rajesh, A. Chandrasekar, "An Efficient Object Oriented Design Model: By Measuring and Prioritizing the Design Metrics of UML Class Diagram with Preeminent Quality Attributes", IJST, Vol.9, No.21, June 2016, pp. 1-9.
2. Hanyong Choi, Sungho Sim, "Design Composition based on DMI architecture", IJAER, Vol.10, No.90, 2015, pp. 681-687.
3. Mario Cervera, Manoli Albert, "Victoria Torres, Vicente Pelechano, A Methodological Framework and Software Infrastructure for the Construction of Software Production Methods", LNCS, 2010,6195, pp. 112-125.
4. Majdi Abdellatief, Abu Bakar Md Sultan, Abdul Azim Abd Ghani, Marzanah A. Jabar, "A mapping study to investigate component-based software system metrics", Journal of Systems and Software, 2013, pp.587-603.
5. Patrick Donohoe, "Software Product Lines: Experience and Research Directions", Springer Science & Business Media, 2012.
6. S. Apel, S. Kolesnikov, J. Liebig, C. Kästner, M. Kuhlemann and T. Leich, "Access Control in Feature-Oriented Programming," Science of Computer Programming, Vol. 77, No.3, 2015, pp.174-187.
7. Mathana, J. M., P. Rekha, V. Sai Anitha, B. Suchitra, and K. B. Bavithra. "VLSI Architecture of MIMO Detector Using Fixed Complexity Sphere Decoding." Review of Information Engineering and Applications 1. 1 (2014): 11-23.
8. Kamboj, S. and Ghumman, N.S. "A Survey on Cloud Computing and Its Types," 2016 3rd International Conference on Computing for Sustainable Global Development, New Delhi, 2016, pp.2971-2974.
9. Arinze, B. and Anandarajan, M. "Adapting Cloud Computing Service Models to Subscriber Requirements", 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2013, Atlantic City, pp. 1-5.
10. Kannadhasan, S., B. Naveen Lingaesh, and R. Alagumanikandan. "Lossless Image Compression and Decompression to Improve the PSNR and MSE Values Using Architecture." Review of Computer Engineering Research 3. 2 (2016): 35-40.
11. Bokhari, M.U., Shallal, Q.M. and Tamandani, Y.K. "Cloud Computing Service Models: A Comparative Study", 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, New Delhi, pp. 890-895.
12. P. Mell and T. Grance. "The NIST Definition of Cloud Computing. NIST Special Publication", September 2011, pp. 800-145.
13. Suryanto, T., Haseeb, M., & Hartani, N. H. (2018). The Correlates of Developing Green Supply Chain Management Practices: Firms Level Analysis in Malaysia. Int. J Sup. Chain. Mgt Vol, 7(5), 316.
14. Salman A. Baset, "Cloud SLAs: present and future", ACM SIGOPS Operating Systems Review, Vol. 46, No. 2, 2012, pp.57-66.