

Variable Service Process for SaaS Application

Nadir K. Salih and Tianyi Zang

School of Computer Science and Engineering, Harbin Institute of Technology, China

Abstract: To meet the required functional and nonfunctional requirements of different enterprise applications it is important to model the possible design so that a feasible alternative can be defined. We observed SaaS developed application by multi-tenancy to provide flexible customization and introduced many issues in software scalability and dynamic testing. Multi-tenancy demand to customizing the single instance according variability wishes among many customers. For that we proposed variable service process to customization multi-tenancy in runtime. It will be realize all benefits of variability concept for SaaS application.

Keywords: Multi-tenancy, SaaS, service process, variability

INTRODUCTION

Variability can be defined as probability to adapt software product to specific situation different platform, specific customer wishes and various location. In software product is needed to offer all specific functionality. It can occur in number of states in design, compilation of software, linking to other software and runtime. Variability concerns affect both the service interface and the service provider implementation and hence are crosscutting in nature (Jegadeesan and Balasubramaniam, 2009). Multi-tenancy demand to customizing the single instance according variability wishes among many customers (Thomas *et al.*, 2008). And this customization can realize by continuous testing (Tsai *et al.*, 2010) for multi-tenancy. A multi-tenant setup will draw more requests and therefore has to be scalable. Scalability in a multi-tenant setup has several dimensions: that is supporting large number of requests, processes and tenants. Related to variability, there are other concerns raised by multi-tenancy. For example, how to ensure correctness of all possible configurations? It is not sufficient to guarantee the correct operation of a single process. Instead one needs to ensure the correctness of a process family and all of its configurations. Another concern is security; How to make sure that data and processes of different tenants are isolated while using the same code base and infrastructure?

Variation management in this context allows the tailored use of services to provide the exact, desired capability for a specific product. The dynamic nature of service invocation may support adaptation and more dynamic growth of a product line scope. SaaS may also support a more opportunistic response to changing market conditions with product line adaptation or new product lines.

We look for make SaaS service management should easily and do proactively, automatically to solve problem (Moving to SaaS for IT service management, 2010). Our contribution in this study how we apply variable service process for multi-Tenancy SaaS. In order to apply dynamic adaptive for SaaS application we proposed variable service process to customization multi-tenancy in runtime.

LITERATURE REVIEW

Service lifecycle: The concept of the Service Lifecycle on the processes required to design, deliver and support services for customers. The ultimate success of service management is indicated by the strength of the relationship between customers and service providers. Service Lifecycle provide the necessary guidance to achieve success by five layers strategic

Objectives, designed, transitioned, supported and improved. It is important to note that most of the processes defined do not get executed within only one lifecycle phase. The primary objectives of Service Strategy are to (SaaS Strategy and Enablement, 2012):

- Design, develop and implement service management a strategic asset and assisting growth of the organization
- Develop the IT organization capability to manage the costs and risks associated with their service portfolios
- Define the strategic objectives of the IT organization

We depict the service lifecycle in Fig. 1 to describe it work.

Service process: SaaS can optimize key service operating and achieve superior customer satisfaction by following steps:

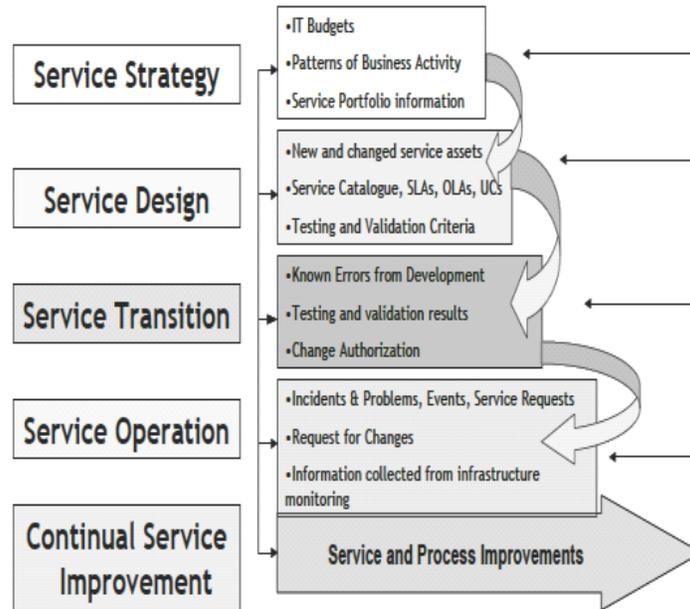


Fig. 1: Service lifecycle work

- Through process build better quality for product
- Better implementation and customer support
- Different structure operational processes for management

The purpose of Service Operation is to achieve stability. Service Operation staff must ensure that changes are absorbed without adverse impact upon the stability of the IT services. There are many things lead to change in services:

- Upgraded in hardware or software
- Changes to meet dynamic business requirements
- Enhancement in process
- Modify in management

Service flexibility: Flexibility service (Zhan and Duan, 2012) is important thing for SaaS because it makes things more scalable and adaptive. The concept of dynamic service routing means that the routing path is not fixed and can be changed by selecting service from the a service set dynamically to meet the need of consumer. An example of some SaaS functionality that might be provided this way might be a workflow mechanism. Acknowledging the fact that a given SaaS vendor product might represent a set of features that are a subset of a larger overall corporate process, it would be useful to provide customers with a simple way to pass in information from systems that are used earlier in the process and then pass data on to systems that are used later in the process. Workflow functionality such as this is being utilized with increasing frequency and SaaS vendors may consider providing services such as this as

part of their overall product offering. Making sure that the implementation of this functionality allows for adequate scaling is critical. The operations side of administration also needs to think about how it will scale as the business grows. It is one thing to keep track of a few dozen customers in a multi-tenancy environment but quite a different exercise to manage a few thousand. Issues such as keeping track of resource usage (including disk space), data backup and restoration and current account status (has the customer been suspended for lack of payment?) are just some of those that can become unwieldy if the administration system is not designed accordingly. It is also important to consider how the provisioning of additional services might be handled.

METHODOLOGY

We can depend on continual service improvement model that can define all requirements of services improvement (Introduction overview from it SMF, 2007) (Fig. 2).

From this model concept we designed a service that can be variable or continuous in improvement at runtime for SaaS. By variable service process at runtime we can make dynamic adaptive for SaaS as depict in Fig. 3.

We look for variable service for two things one requirement change, second satisfy customer to level service. From Fig. 3 Service level management can be responsible from designing, determine the level service requirement and realized service level agreement. It works after monitor state to result improvement in service. In addition it can make analysis for SLA to understand the change in user requirements. We can

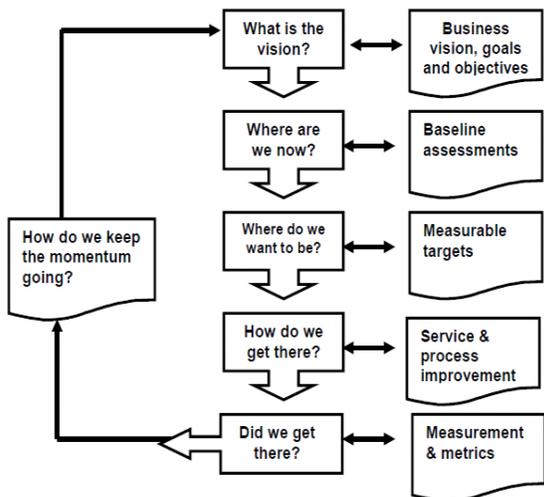


Fig. 2: Continual service improvement

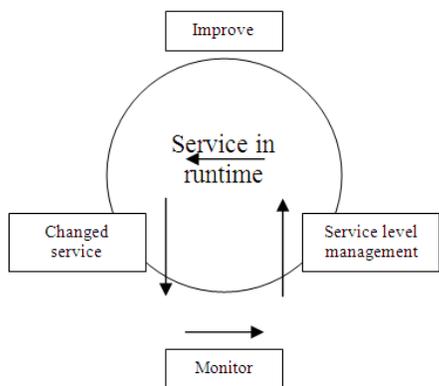


Fig. 3: Variable service process

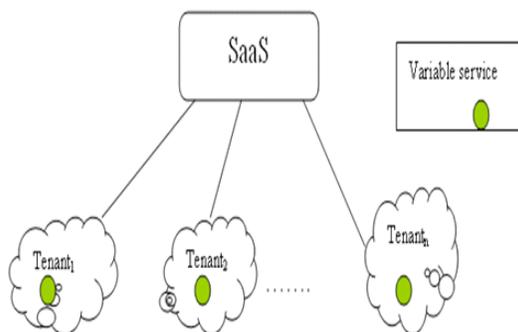


Fig. 4: Multi-tenancy with variable service

measure some parameters service level achievement, cost of services and number of request from customer by measure of this parameter we can reflect dynamically change to our service. This service variation we applied it in multi-tenancy SaaS application for easy customization user requirements change Fig. 4.

This variability in service process will help any tenant to become dynamic adaptive. And it can be self adaptive in progress of research. This will be starting point to apply autonomously management to SaaS environment.

DISCUSSION

Ghaddar *et al.* (2011) they reduced the complexity in SaaS by made variable mode and manage any application layer with variation. Mietzner *et al.* (2009) they managing the variability in SaaS application by variability model techniques. Schroeter *et al.* (2012) they identified runtime architecture to dynamically adaptive SaaS. Dong and Ku (2009) they mentioned the variability and commonality can be maximizing reusability. Kabbedijk and Jansen (2011) they propose three architecture patterns to manage multi-tenancy. Mietzner and Leymann (2008) Used variability in process layer to allow different customizations for SaaS. Mietzner *et al.* (2008) show how the Service Component Architecture (SCA) can be extended with variability descriptors and SaaS multi-tenancy patterns to package and deploy multitenant aware configurable composite SaaS applications. All this related work used variability concept in architecture but we applied it in service process. From our proposed we can facilitate service process customization at runtime and the composition of SaaS service.

CONCLUSION AND RECOMMENDATIONS

This study, we discussed development in cloud/SaaS exactly in multi-tenancy. We showed that supporting variability is one of the main challenges. Moreover, we discussed the potential of variable service process techniques in such environments. We believe that configurable services process in the cloud enable a new kind of progress. Prominently in multi-tenancy because service provider need to save a time and money and give quality of service for customer. For that variability concept is very important. Ongoing study will be apply this variability in runtime and make dynamic adaptive for usage resources in realwork.

ACKNOWLEDGMENT

This study has been developed with the support under the project with number: 2012AA02A604, 863 Program key projects in China: The Technology and the System Development for Smart Acquirement of Personal Healthcare Information. And so the Key Project of NSF in China: Methodology of Value-oriented Software Services: Theory, Method and Application with number: 61033005.

REFERENCES

- Dong, S. and D. Ku, 2009. A systematic process for developing high quality saas cloud services. Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09, Springer-Verlag, Berlin, pp: 278-289.
- Ghaddar, A., D. Tamzalit and A. Assaf, 2011. Decoupling variability management in multi-tenant SaaS applications. IEEE, International Symposium on Service Oriented System Engineering, pp: 273-279.
- Introduction overview from it SMF, 2007. Retrieved from: <http://www.ucisa.ac.uk/en/representation/activities/ITIL>.
- Jegadeesan, H. and S. Balasubramaniam, 2009. A method to support variability of enterprise services on the cloud. IEEE International Conference on Cloud Computing, pp: 117-124.
- Kabbedijk, J. and S. Jansen, 2011. Variability in multi-tenant environments: Architectural design patterns from industry. Springer, 6999: 151-160.
- Mietzner, R. and F. Leymann, 2008. Generation of BPEL customization processes for saas applications from variability descriptors. IEEE Int. Conf. Serv. Comput., 2: 359-366.
- Mietzner, R., F. Leymann and M.P. Papazoglou, 2008. Defining composite configurable saas application packages using SCA. Third International Conference on Internet and Web Applications and Services, ICIW '08, 8-13 June, Stuttgart Univ., Stuttgart, pp: 156-161.
- Mietzner, R., A. Metzger and F. Leymann, 2009. Variability modeling to support customization and deployment of multi-tenant-aware software as service applications. Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, PESOS '09, Washington, DC, USA, pp: 18-25.
- Moving to SaaS for IT Service Management, 2010. Retrieved from: <http://www.expresscomputeronline.com/20100614/techviews01.shtml>.
- SaaS Strategy and Enablement, 2012. Retrieved from: <http://www.infostretch.com/SaaS/saas-strategy-and-enablement.php>.
- Schroeter, J., S. Cech, S. Götz, C. Wilke and U. Aßmann, 2012. Towards modeling a variable architecture for multi-tenant saas-applications. Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, VaMoS '12, NY, USA, pp: 111-120.
- Thomas, K., N. Thao and L. Linh, 2008. A software as a service with multi-tenancy support for an electronic contract management application. IEEE Int. Conf. Serv. Comput., 2: 179-186.
- Tsai, W., Q. Shao, Y. Huang and X. Bai, 2010. Towards a scalable and robust multi-tenancy SaaS. Proceedings of the Second Asia-Pacific Symposium on Internetware, Internetware '10, NY, USA, Article No. 8.
- Zhan, Y. and L. Duan, 2012. Method of flexible thought assembling saas service. Springer, 137: 807-814.