Change Management Based on Eccentricity Factor

V.Kavitha[1], P. Shanmugapriya[2]

[1]*Associate Professor, Department of Computer Science and Engineering, Sri Sai Ram Engineering College, Chennai , Research Scholar, SCSVMV University, ,*

[2]*Assoc. Prof, Department of Computer Science and Engineering, SCSVMV University.*

Abstract **-** Version control systems are employed for change management on the existing code base during development phase and maintenance phase. The changes may or may not affect the existing ones. If the changes affect the existing code they may lead to errors (compile time and run time) or instability of the system. Version control systems employ various strategies to cater the needs of various scenarios but it leads to delays. The agility of the system depends on the updating mechanism and the strategy it deploys while handling multiple users and dependent modules at the same time. The ability to handle changes by the version control systems depends upon the strategy to organize the current modules, update them and serve them to multiple users at the same time. The objective is to propose a system that is found to be an enhanced version of the current system.

Keywords- Version control systems, change management, instability, multiple users, dependent modules, enhanced version.

I. INTRODUCTION

In order to be competitive in the environment of commerce globalization, realizing inter-organizational cooperation and association becomes an inevitable trend for nowadays organizations. As a result, these organization often put their industry processes into a distributed and dynamic environment with the help of Service-Oriented Computing (SOC) and Web Service (WS) technologies. There are a large number of organizations that interact with each other by providing and invoking services [10]. An organization can act as a service provider, service consumer or both of them and this kind of business processes is referred to as service-based business processes (SBPs). Growing competition, financial stress, and customer anxiety are causing service providers to transform the way they create and deliver services. The vital goal of this transformation is to achieve faster time-to-market, leverage shared enablers across applications, achieve principal and operational savings, and generate additional subscriber and non-subscriber revenues from non-traditional areas, such as advertising. Through-out this transformation, service providers are keeping an eye on the desires of end users [1]. Due to the distributed and vibrant nature, changes of the internal business process and involved services of an SBP can happen often. There are dependencies between the internal business process and involved services [7], therefore, the changes occurring in one side may affect the other one in certain degrees and can propagate in the whole SBP like the ripple effect which makes the change management in SBP a really challenging problem [18]. Present Business process is technology intensive, so dealing with composite technology challenges is not new. There has been significant progress in implementing internal and inter-organization IT solutions that provide better framework for business decisions which reduces administrative costs and improve customer satisfaction by reducing errors. In this scope, we will concentrate on potentials of Web services to construct a full process from stem to stern with extraordinary concentration on the business aspects. Moving a higher percentage of business process outside the industry is necessary and significant component of strategic change described. Change management is considered one of the demanding problems in the field of software engineering, database and workflow systems. In [6] [7], some works deals with the changes in highly tightly-coupled system have been proposed. A monitoring module is usually deployed to handle, forward and self-adapt to the occurrence of changes. Due to known drawbacks of central approach, this work will not consider a central mechanism for change management.

As the composed/aggregated services available increases, the complexity of managing changes will raise and the manual management of changes becomes difficult or even impossible. One of the greatest promises of the distributed business process environment is the capability to self-adapting to guarantee goals achieving.

Abstracting changes in business relationships requires an automated approach to manage changes without any blow on high level functionalities offered to users. For example, the proposed hotel booking environment must be able to automatically plug-in/plug in-out Web services with little overhead while guaranteeing both *functional* and *non-functional* properties. Functional properties refer to the functionalities that the business process environment have to accomplish. The non-functional properties refer to events surrounding the functional properties. Change management is therefore a critical component in the deployment of any business process environment. Our works provides such as framework for change management based on eccentricity factor. To ensure desired requirements, different practical approach can be used. An example of these approaches is Bottom-up approach in which the changes are initiated in the Web service environment. For example, a service may become unavailable during execution and triggers dependencies services, then user has to replace this service.

A small change might have a ripple effect on the other activities which may put down the stability of the entire structure. Making changes to the existing code base is difficult during the following situations: when there are live users to be

mitigated, when there are multiple developers working on the same product feature, when there is a highly-coupled product. Nowadays, version control systems are good enough to cater these ambiguities. The rate at which the version control system serves the developers can be improved. The dependencies are estimated by the VCS and other tools by parsing each and every part of the program, thereby building a network based on flow of data to find out most dependent module to begin with. The data flow is manipulated based on stacking of the modules and the one on the top is the most dependent module. A super repository, sub repository maintenance also facilitates the change management. When someone works (Write mode) on the most dependent module: a lock is established to restrict others using the same module, a lock is established to restrict others using the dependent modules, wait queue is maintained to give priorities for critical tasks and future access to the same module and dependent modules. The system takes some time to update the changes and re-determine the network flow. The users must additionally wait for this process to complete.

II. IMPLEMENTATION

Let's look at the total time estimate of the current system under working. Let's label the user write time as 'X' and update time as 'Y'. 'X' and 'Y' are dynamic variables i.e., they need not be same in all cases. 'X' and 'Y' may depend upon the person, type of module and various other factors. Now the next developer in queue must wait 'X+Y'. One must also consider 'interrupts' which will lead to reordering of the queue. Let's label the interrupt factor to contribute to 'Z'. 'Z' is again a dynamic variable. Now the total time the next developer must wait is 'X+Y+Z'. The goal is to enhance the updating process i.e. 'Y'. The new consideration also proposes a significant reduction in 'X' and 'Z' when looked as a whole. The entire network of the program can be updated easily by considering it as a graph. The eccentricity plays a vital role in the same. "Eccentricity is the measure of the farthest distance from that particular node to the terminal nodes". The centre has the least eccentricity. Every graph has 1 or 2 centres which can be easily determined by the eccentricity factor. Once it is determined, the centres are the most dependent modules in the whole program. The modules can be subject to changes one by one from centre to the terminals based on the eccentricity factor. This makes it easy for the programmer and also the updating system. We can expect a significant change in the efficiency towards the positive scale. The updating process of new eccentricities is estimated to be quicker compared to the traditional system. The proposal can be termed as 'X+Y+Z-α' where 'α' is the significant improvement. The improvement varies with various factors such as the system capacity, user efficiency, network efficiency, etc.

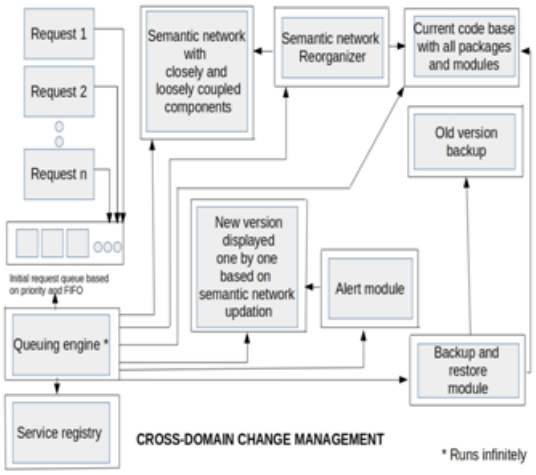

FIG. 1 Semantic network is completely built and maintained on eccentricity factor

The requests are put into a priority based queue that works on FIFO strategy with interrupt handling embedded into it. The service registry is referred into if there is a cross domain – change requirement. A semantic network based on eccentricity is built and maintained which displays the modules (one or two) at the centre initially (the ones with the least eccentricity) followed by the others (based on increasing eccentricity). We consider trees here – Reason being that developers don't intentionally create deadlocking modules. If there are any they can be found by the threshold being maintained for each module. A maintenance module reorganizes each and every part of the network based on the dynamic programming approach. A backup and restore module also maintains the previous versions that can be rolled back into. Locks are performed when necessary as done in current Version control systems. Locks are established when there are multiple developers making changes on a module and its dependent modules are also locked in certain situations.

III. FINDINGS

The dynamic programming approach based on eccentricity gives a cutting edge in the entire process thereby improving the process. It is also proposed that if AI is coupled along with this approach the results would be much better as AI reduces human error and reduces 'X+Z' factor as mentioned above thereby improving 'α'. This there by reduces the waiting time for the developers to access the module under lock and dependent modules under lock.

IV. CONCLUSION

The overall objective of the proposal is to enhance the delay and the smoothness of the cross – domain change management. The proposal doesn't reduce the necessity for lock instead reduces the waiting period.

V. REFERENCES

[1]. Rey, S. and Sarrazin, H. How telecoms can get more from Internet Protocol. McKinsey on IT, 2006.

[2]. Haaker, T., Kijl, B., Galli, L., Killström, U., Imonen, O. and de Reuver, M. Challenges in designing viable business models for context-aware mobile services. 3rd International CICT Conference (Mobile and Wireless Content, Services and Networks, 2006.

[3]. Brenner, M., Grech, M. and Unmehopa, M. The service delivery Environment a standards-driven melting pot. International Conference on Intelligence in Next Generation Network, 2007, 8-16.

[4]. N. Crespi. How telecoms can get more from Internet Protocol. Sciences Technology Information projects, 2009.

[5]. Park, K.H. and Favrel, J. Virtual enterprise-Information system and networking solution. Computers & industrial Engineering **37** (1-2) (1999) 441-444.

[6]. Madhavji, N.H. Environment evolution: The prism model of changes. IEEE Trans. Software Engineering **18** (5) (1992) 380-392.

[7]. Cobena, G., Abiteboul, S. and Marian, A. Detecting changes in xml documents. Proceedings of the 18th International Conference on Data Engineering, 2002, 41-52.

[8]. Aalst, W.M.P.V.D The application of Petri Nets to workflow management. Journal of Circuits, Systems, and Computers **8** (1) (1998) 21-66.

[9]. Mtibaa, S. and Tagina, M. A Combined Petri Nets and Model-Driven Engineering for Requirements Specification Approach to Analysis of Service-Based Applications. un-published.

[10]. Akram, M.S. Managing Changes to Service Oriented Enterprise. PhD dissertation Faculty of the Virginia Polytechnic Institute and State University, 2005.

[11]. Brock, M. and Goscinski, A. State aware WSDL. Proc. Sixth Australasian Workshop on Grid Computing and e-Research, 2008, 82 (7) 35-44.

[12]. Hinz, S., Schmidt, K. and Stahl, C. Transforming BPEL to Petri Nets. Proceedings of the Third International Conference on Business Process Management, 2005.

[13]. Badouel, E., Llorens, M. and Olivier, J. Modeling Concurrent Systems: Reconfigurable Nets. Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications, 2003, 1568-1574.

[14]. Brambilla, M., Ceri, S., Comai, S. and Tziviskou, C. Exception handling in workflow driven web applications. Proceedings of the 14th international conference on World Wide Web, 2005, 170-179.

[15]. Rinderle, S., Reichert, M. and Dadam, P. On dealing with structural conflicts between process type and instance changes. International Conference on Business Process Management, 2004, 274-289.

V.KAVITHA, Assoc.Prof., Dept. of computer science and Engineering, Doing her Research in the domain of Service Oriented Architecture. Published papers in various national , International and scopus indexed journals.

Dr.P.SHANMUGAPRIYA Assoc.Prof., Dept. of computer science and Engineering, SCSVMV University. Her area of interest is Software Engg. Published papers in various International Journals.