EVALUATION MODEL FOR WEB SERVICE CHANGE MANAGEMENT BASED ON BUSINESS POLICY ENFORCEMENT

G.Nandhini[1],P.Inbavalli[2],T.Halima Begum[3]
PG Scholars
Department Of Computer Science And Engineering,Pondicherry University
Sri Manakula Vinayagar Engineering College, Puducherry,India.
nandhini1171992@gmail.com
inbacse90@gmail.com
halimafarz@gmail.com

***Abstract*-Today organizations in all industries are increasingly dependent upon IT and a highly available network to meet their business goals. As customer expectations is one of the current problems faced by IT enterprises at present, leading to increasingly complex IT service management systems. Although there exists many change management issues and solutions, there is no proper support for efficient change evaluation and monitoring. Present change management works are done only in the web service interface such as WSDL Web Service Definition Language and there is no proper support for dynamic nature in them. In order to provide a flexible environment for the Business Analyst to perform the emergency changes over the service business logic without the need of IT staff and also to evaluate changes made, we propose a Finite State Machine based Evaluation Model for Web Service Runtime Change Management and also a set of change factors to evaluate runtime changes of web services. Among which we mainly focus on the factor Business Policy Enforcement which is used for checking whether there is any policy violation with respect to the changes made. We first define the various change factors and derived solution is considered as the change measure for individual change factor.**

***Keywords*-Web Services, BL Analyzer, Dependency Analyzer, Change Factors, Business Policy Manager, Runtime Change Management, Change Evaluation Report.**

I. INTRODUCTION

A web service is a piece of business logic that enables a paradigm shift in business structures allowing them to outsource required functionality from third party Web-based providers. As services are not static, they undergo many changes during their life cycle since the IT service management system now is facing an increasingly complex environment aroused by wide variety of requirements. This situation increases the likelihood of service interruption which in turn causes business loss. In this case, a series of changes are needed so that potential service interruptions can be prevented. A typical IT enterprise faces considerable number of changes each year. To handle such changes, change management is needed.

Change management is a set of processes that are employed to ensure that significant changes are implemented to a business process during its maintenance phase. The purpose of the change management process is to ensure that Business risk is managed and minimized; authorized changes support business needs and goals. Hence there arises the need for an effective framework managing those changes without affecting the business functionality. Change Management framework is proposed for making minor alterations to business logic but whose effect is more pronounced to the profitability of the organization. So the changes in business logic of the web service must be done very quickly within the given time constraint as the services consumed by the providers and clients must not be affected there by managing the changes at the runtime.

A specific change in the logic, not only changes the functionality concerned with it, but also the dependent functions under the dependency group. Those dependency groups assist in performing changes and results in an effective change. Those changes that are made based on the dependency analysis are measurable, complete and finite. Any change that is not based on the dependency analysis is said to violate the outlined properties and may result in a failure. Dependency analysis based on each factor is performed by a finite state machine. With the help of finite state machine, it is possible that if a particular logic has been successfully recognized then it goes to the halt state. On the other hand, there exists a transition from any state to the exception state E, if the rule or function encounters an exceptional input or parameter. By this way, the FSM simulates the pattern based on the behavior of the rules and functions, which can be easily verified and measured at each stage. This measure of change evaluation ensures that the business analyst has a direct control over the changes he makes and provides an environment that notifies the analyst about the outcome in a meaningful way. We propose a set of five change factors based on which the dependency existing between the business rules, functions and parameters is analyzed. Among which we mainly focus on the factor Business Policy Enforcement which is used for checking whether there is any policy violation with respect to the changes made. These change factors are also involved in the change evaluation that makes the changes to be specified in a precise and formal manner.

II. FRAMEWORK FOR WEB SERVICE CHANGE MANAGEMENT

In this framework, the course of action begins with issue of change request and the whole processing of the request is conducted under the supervision of an analyst. A Change Request is an 'additional' expectation that the Customer wants which is outside the current defined scope of the project which is a change to an existing product. It is first specified by the analyst in detail. This specification includes the change to be made, brief description of the change to be made, action to be made, services, rules and the conditions under which the change has to be made. The Request handler is responsible for finding the type of the change and priority is assigned to indicate the order of importance of the request and also determines whether the change can be implemented within the runtime. Domain Variable Identifier discovers the domain variables present in the request by just tokenizing the request and searching whether any of tokens match against given domain variable set. Once we are done with domain variable identification, we can determine the domain to which the request refers to. The Domain Mapper finds the corresponding domain and services with the help of service repository which comprises

all kinds of services and provides the result to Business Logic Analyzer.

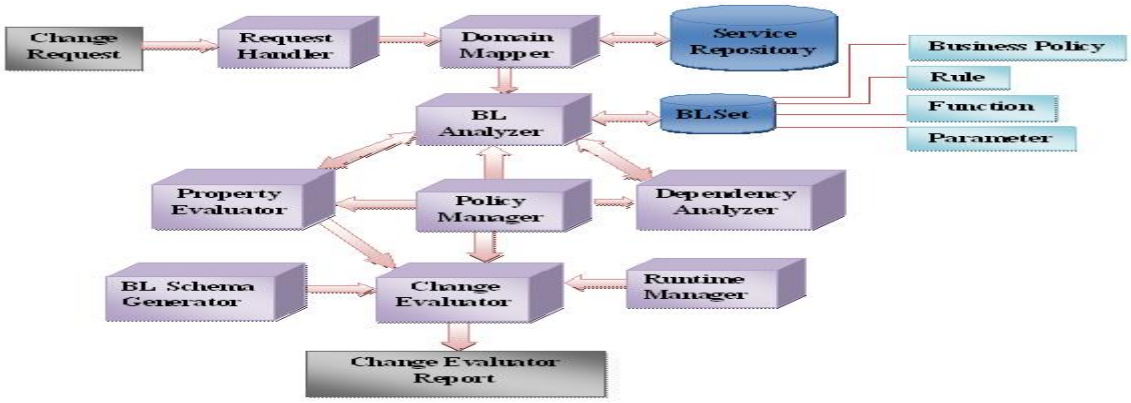

Fig1.Change management framewor

The logic of all the services in the service repository are decomposed into rules, functions and parameters along with associated business policies are stored in the BL set with help of domain mapping. The BL analyzer identifies only the particular rule, functions, parameter and policy associated with it for each change request from BL set instead providing all the rules, functions, parameters and policies associated with it. The properties of interoperability, traceability, decidability, computability are preserved even after the alteration. These properties are evaluated by property evaluator and the calculated properties are appended to the schema to generate the final schema. The property values assist the analyst in making a decision whether or not to implement the alteration. Dependency Analyzer discovers dependencies with the ascertained rules and functions in the business logic through the identified change criteria. The Policy Manager checks for policy violation in the rule or function in which change has to be made and also in the dependent rules or functions. Since the change which is complete, finite and computable can even violate the associated policies, it is also checked by the policy manager after evaluating the properties of the identified change. Schema Generator is responsible for generating and upholding BL schema at various stages. This BL schema is helpful in guiding the management people who are not familiar in source code to make fruitful changes. The changes made in the schema will be updated obviously in business logic. Same way, when programmers do some changes in logic, it gets updated in schema accurately. Run time manager is responsible for building and deploying the services after fruitful changes are done and also it helps to debug the exceptions in the service logic sophistically. Change evaluator evaluates the changes made by the analyst based on the change factors such as business policy enforcement which determines violation of policy in the identified change criteria, code consistency which checks whether the preconditions and post conditions remains constant with respect to the changes made, requisite measure which checks whether the requisite rule or function is not altered with respect to the changes made and mapping function which checks the consistency with respect to the logic and also on the non-functional factors such as service interruption time, response time, reliability and service availability. The Change Evaluation Report includes result of the evaluation process of the change made i.e. the evaluation measures of the changes made by the analyst. By this framework, the analyst has the advantages such flexibility, works in a sophisticated environment without having much knowledge about source code since he can implement the changes at the schema level, knowledge transfer since he can do the changes whatever comes in mind directly and is able to evaluate it easily, independency since no need for depending on the development team thereby reducing the time and cost.

III. CHANGE FACTORS

Change factors are those that act as a criteria based on which the changes made can be evaluated. Any change in the business logic is governed by the following change factors. These factors influence changes either independently or in a group and make changes effective. And the change factors are as follows:

- Business policy enforcement
- Requisite measure
- Code consistency
- Mapping function

A. Business Policy Enforcement

In this paper, we mainly focus on the change factor business policy enforcement which is a measure of change which determines violation of policy in the identified change criteria. Here the main role of business policy enforcement is: policy mapping which is nothing but finding the appropriate policies which are associated with the rule or function in which the change has to be made and also with its dependent rules and functions; validation which refers checking whether there is any policy violation; and refinement which provides an environment for the analyst to modify the policy itself if necessary.

TABLE I

EVALUATION RESULT FOR BUSINESS POLICY ENFORCEMENT

Change Request	Business Rule	Business Policy	Evaluation Result		
			Policy Partially Denied	Policy Fully Denied	Policy Accepted
Customers with military as occupation wants to do transaction at any time till null balance for all his/her accounts	For all customers except customers with occupation as military, Minimum balance = 500	Customers with military as occupation can be allowed to do transaction till null balance for only one account.	TRUE	-	-
Any person in joint account can remove the other persons sharing the joint account at any time without their permission and can create a separate account.	Two or more persons can share a joint account (i.e. two or more persons can have the same account name)	A person in joint account can remove the secondary person without his/ her permission if the former is the only owner.	-	TRUE	-
Request to change the minimum balance for all the accountholders	For all customers except customers with occupation as military, Minimum balance = 400	Military accountholders should have minimum balance as null	-	-	TRUE
Changing the maximum amount that can be transferred between two accounts per day	Maximum amount transferred should be 20000 per day	Maximum amount transferred should be greater than minimum balance.	-	-	TRUE
certain amount of tax must be added to all kinds of loans	tax percentage provided for each kind of loans	No tax must be paid for educational loans	TRUE	-	-

In a business logic L encompassing set of rules R, functions F, parameters Pr, policy set P and dependency D , the change which is going to be made can be evaluated based on the business policy enforcement factor which checks whether there is any policy violation with respect to the changes made. Whenever a change is specified, it is first analyzed for completeness, finiteness and computable. Once the rules, functions and parameters in the change specifications are analyzed as complete, they are mapped with the existing logic set L. Then the corresponding rules, functions, parameters, dependency set and policy set are retrieved by which the change specification is checked whether it violates any policy. If so, the analyst is informed as the change cannot be made due to the violation of policy. Otherwise the change is successfully included in the existing logic set L.

IV. EXPERIMENTAL STUDY

The table (TABLE I) shown above is the evaluation results of some of the change requests implemented based on the business policy enforcement. Below Fig 2 and 3 shows overall performance and average bandwidth of business policy are evaluated and any type of change request can be handled very easily by business analyst.

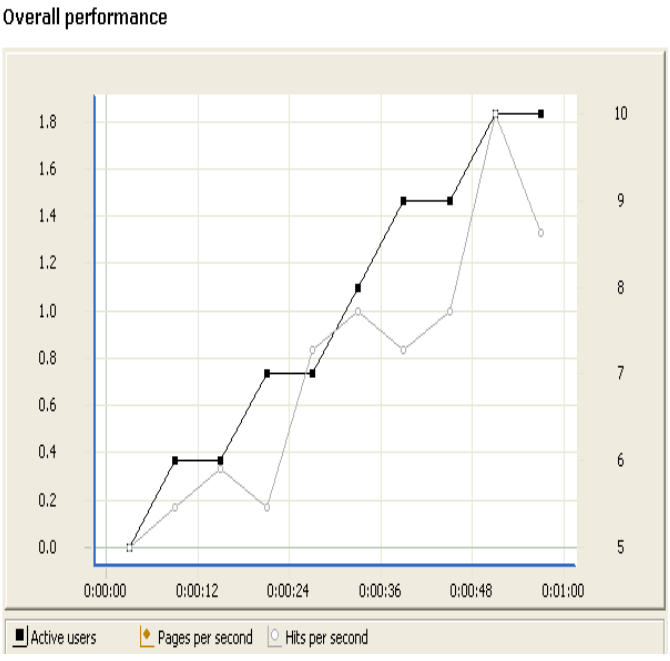

Fig 2. Overall performance

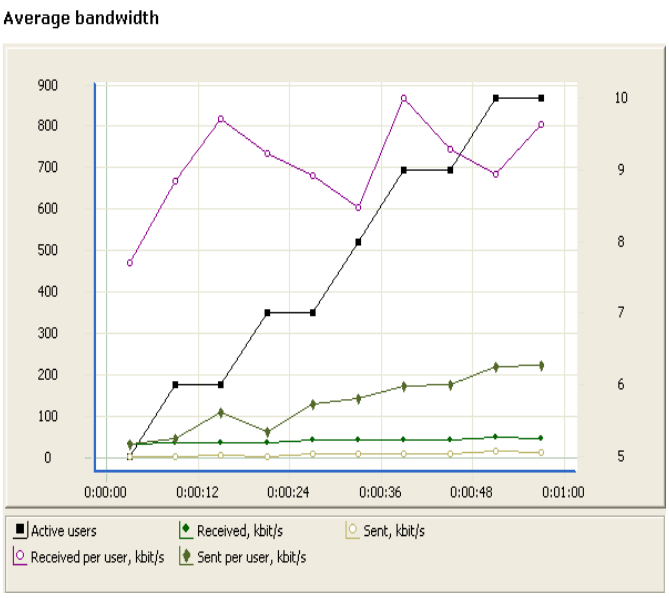

Fig 3. Average Bandwidth

V. CONCLUSION

Any organization no matter its size will experience a large volume of changes in order to accommodate new business requirements, to correct faults in the infrastructure or the services, or for other reasons (such as legal requirements). This framework provides an environment for the business analyst to implement the changes at the sophisticated environment without having much knowledge about the source code and to evaluate the changes easily based on the proposed change factors in which business policy enforcement is mainly focused. By this framework, analyst is able to implement whatever comes in his mind directly without the help of developing team and also to detect policy violation and do the changes accordingly and also to refine the policy itself if needed.

REFERENCES

[1] Felix Cuadrado, Juan C. Duenas and Rodrigo Garcı´a-Carmona, "An Autonomous Engine for Services Configuration and Deployment," *IEEE Transactions on Software Engineering*, Vol. 38, No. 3, May/June 2012.

[2] Xumin Liu, Athman Bouguettaya, Jemma Wu, and Li Zhou, "Ev-Lcs: A System For The Evolution Of Long-Term Composed Services," *IEEE Transactions on Services Computing*, Vol. PP, Issue 99, 2011.

[3] Dimitris Apostolou , Gregoris Mentzas,Ljiljana Stojanovic, Barbara Thoenssen, Tomás Pariente Lobo,"A collaborative decision framework for managing changes in e-Government services," *Government Information Quarterly 28 (2011) 101–116, PUBLISHED @ ELSEVIER*, 2011.

[4] Bruno Wassermann, Heiko Ludwig, Jim Laredo, Kamal Bhattacharya, Liliana Pasquale, "Distributed Cross-Domain Change Management," *2009 IEEE International Conference on Web, Services*, 978-0-7695-3709-2/09, 2009 IEEE.

[5] Bassam Atieh Rajabi, Sai Peck Lee, "Change Management in Business Process Modeling Survey", *2009* International *Conference on Information Management and Engineering*, 978-0-7695-1/09, 2009 IEEE.

[6] Bassam Atieh Rajabi, Sai Peck Lee, "Runtime Change Management Based on Object Oriented Petri Net," *2009 International Conference on Information Management and Engineering*, 978-0-7695-1/09,2009 IEEE.

[7] Oliver Moser, Florian Rosenberg and Schahram Dustdar,"Domain-Specific Service Selection for Composite Services," *IEEE Transactions on Software Engineering*, VOL. 38, NO. 4, July/Aug. 2012

[8] Ruhi Sarikaya, Canturk Isci and Alper Buyuktosunoglu, "Runtime Application Behavior Prediction Using a Statistical Metric Model," *IEEE Transactions on Computers*, Vol. X, No. X, May 2011.

[9]Haochen Li, Zhiqiang Zhan, "Bussiness-Driven Automatic IT Change Management Based on Machine Learning,"978-1-4673-0269-2/12, 2012 IEEE.

[10] Yanhua Du, Xitong Li, and PengCheng Xiong, "Petri Net Approach to Mediation-Aided Composition of Web Services," *IEEE Tansactions on Automation Science and Engineering*, Vol. 9, No. 2, Apr. 2012.

[11] Marcello La Rosa, Arthur H. M. ter Hofstede, Petia Wohed, Hajo A. Reijers, Jan Mendling, and Wil M. P. van der Aalst, "Managing Process Model Complexity via Concrete Syntax Modifications," *IEEE Transactions on Industrial Informatics*, Vol. 7, No. 2, May 2011.

