ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

A Finite State Machine based Evaluation of Business Policy Enforcement in Long Term Composed Services

M. Thirumaran[1], M. Jannani[2]

Assistant Professor, Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India[1]

PG Scholar, Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India[2]

ABSTRACT: Web services are dynamically evolving entities which provide machine to machine interoperable interaction over a network. With the evolution of Service Oriented Enterprises (SOEs) and dynamic business environment, Web services have attained greater attraction. Change management in Web services in long term composition involves analyzing the change request and the service and incorporating the change cautiously without any issues. Though Web service change management is a wide area of research, there are no standard methodologies adopted for the evaluation of the change. In this paper, we focus on the evaluation of the changes made with the Finite State Machine as the methodology. Business Policy Enforcement in LCS is evaluated by policy mapping using Finite State Machine and is elucidated using Passport system as the case study.

KEYWORDS: Service Oriented Architecture, Web Services, Change Management, Business Policy, Long term Composed Services

I. INTRODUCTION

Web services provide a new approach for accessing systems in a loosely coupled, platform independent and standardized manner. Though Web services are an evolutionary step beyond component based software architectures, they do not escape the necessity of the software lifecycle and one of its most challenging aspects: change management. On the contrary, Web services impose an even bigger change management challenge than does component based software. Currently there are no sound mechanisms and engineering principles for changing Web services and their related entities.

Change management is a structured approach for ensuring that changes are thoroughly and smoothly implemented, and that the lasting benefits of change are achieved. The focus is on the wider impacts of change, particularly on people and how they, as individuals and teams, move from the current situation to the new one. The change in question could range from a simple process change, to major changes in policy or strategy needed if the organization is to achieve its potential.

With the new paradigm of Service Oriented Computing, many enterprises attempt to utilize services as fundamental elements for developing applications as an additive layer on top of existing components and hence services in long term composition (LCS) have drawn greater importance as each service contributes towards the attainment of the business goal with its competency. So the paper focuses on evaluation of the changes made in long term composed services by detecting the policy violations if any from the unchanged service. Finite State Machine is the methodology used for the evaluation of the changes and hence the proposed change evaluation is based on structured and standard approach. This enables efficient evaluation of business policy enforcement through policy mapping.

II. RELATED WORK

The works related to the focus of the research are discussed in this section. An automatic change management framework that is based on the Petri net models has been proposed [1]. Mapping rules and propagation algorithms that handle the triggering changes and reactive changes respectively have been proposed. A change enactment strategy that actually implements the changes and a prototype system for the Ev-LCS has been proposed to demonstrate its effectiveness [2]. A change management framework has been proposed for service oriented enterprises which focuses on bottom up changes [3]. A cognitive approach to business process management elucidating from process logic to business logic has been proposed [4]. A dependency impact analysis model for evolution of Web services which analyses the impact of the dependencies has been proposed [5]. A framework supporting change impact analysis for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Service Oriented Business Applications has been designed [6]. The concept of Context-adaptable Web service policies has been elucidated [7].

III. PROPOSED METHODOLOGY

Business policy enforcement 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.

Case Study:

In a Passport system, consider a change request with the description that appointment scheduled more than thrice, and then no more appointment can be schedule. It will create new account for the applicant

R1 (Applicant Account Creation)

F1-To get the Applicant Account Creation first we have to get the personal details such as their name, age, current & permanent address, phone_ no, email_id, password, Applicant_id, Father's name, Educational qualification, Type of passport, ECR status. P1 is the Applicant name who has applied for the passport. P2 is the age of the Applicant for the minor or major status. P3 is the current address and P4 is the permanent address of the Applicant to whom the information is sent. P5 is the phone_no of the Applicant to give any information. P6, P7 is the email_id,password of the Applicant for the valid Account creation. P8 is the Applicant_id whether it is valid or not. P9 is the Father's of the Applicant to check whether information is valid or not. P10 is the type of passport whether it is normal, tatkaal, fresh or reissue. P11, P12 is the Educational Qualification and ECR status for the verification. P13 is the Marital status of the Applicant to check whether he/she is single or not. There is no dependency for F1.

F2 - To get the Applicant's details and create a valid account for the Applicant. So F2 is dependent on F1.

R2 (Applicant Validity)

F3- After creating the Applicant's account, it has to be checked whether the account is valid or not. So F3 is dependent on F2.

R3 (Eligibility)

F4- It is used to check whether he/she is the citizen of respecting country. P14 is the nationality of the Applicant to check, whether he/she is Indian or another citizen. P15 is the city of the Applicant. So F4 is dependent on F3.

F5- It is used to check age for the major or minor status. P16 is the birth place to check whether it is valid or not. P17 is the Parent's citizenship; if the Applicant is minor he/she need to provide parent's citizenship for the passport. So F5 is dependent on F3.

F6- It is used to check Identification is valid or not. P18 is the photo of the Applicant to whom passport required. P19 is the ID card of the Applicant to check he/she is the same person or not. So F6 is dependent on F3.

F7- It is used to validate other proofs. P20 is the PAN card, Driving License etc. So F7 is dependent on F3.

R4 (Passport Approval)

F8- After the documents verification, the appointment is schedule to the Applicant. On which date the applicant should be held is P21, on what time the applicant is held is P22, at which city and centre the applicant is held P23, P24 and the appointment_no is P25. So F8 is dependent on F3, F4, F5, F6, and F7.

F9- After the appointment, the applicant status is checked. Applicant_id is given to find out whether the Application is processing, rejected or approved is P26, P27, P28. So F9 is dependent on F8.

F10- Finally passport status is found out whether passport is pending or dispatched is P29, P30. So F10 is dependent on F9.

POLICIES:

PO1: If the Applicant is a minor provide Parent's/Guardian details, PO2: If the type of passport is re-issue, details of old passport needs to be provided, PO3: If ECR status is "No", then provide proof of education, PO4: If Appointment

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

scheduled more than thrice, then no more appointment can be scheduled, PO5: Age, Identity, citizenship proof should be a document from the provided list.

FINITE STATE MACHINE REPRESENTATION

Here the state represents rule and transition is represented by using the symbol δi which includes the current state and the input which may also be an internal transition. Within the internal transition, the state is the function and similarly its transition includes current state and input which includes parameter set, policy set and dependency set. Each state has an exceptional state which decides whether that state can be rolled back or not.

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.

Table 1: Business Logic Set comprising of rules, functions, parameters, policies and dependencies

Rules		Functions		Parameters		Policy	Dependency
Applicant Account Creation	R1	get()	F1	name	P1	P02	NULL
				age	P2		
				Current address	P3		
				Permanent address	P4		
				Phone no	P5		
				Email_id	P6		
				password	P7		
				App_id	P8		
				Father's name	P9		
				Type of passport	P10		
				Educational qualification	P11		
				ECR status	P12		
				Marital status	P13		
		create()	F2	name	P1	NULL	F2->F1
				Age	P2		
				Current address	P3		
				Permanent address	P4		
				Phone_no	P5		
				Email_id	P6		
				Password	P7		
				App_id	P8		
				Father's name	P9		
				Type of passport	P10		
				Educational qualification	P11		
				ECR status	P12		
				Marital status	P13		

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Applicant Validity	RR2	Valid account()	F3	App_id	P8	NULL	F3->F2
				Name	P1		
				Age	P2		
				address	P4		
Eligibility	R3	Validate citizenship()	F4	Name	P1	P05	F4->F3
				Age	P2		
				Address	P4		
				Nationality	P14		
				city	P15		
		Validate Age()	F5	Name	P1	P01,P05	F5->F3
				Age	P2		
				Birth place	P16		
				Parent's citizenship	P17		
		Validate identity()	F6	Photo	P18	P05	F6->F3
				ID card	P19		
		Validate other()	F7	Other proofs	P20	P02,P03	F7->F3
Passport Approval	R4	Appointment()	F8	App_id	P8	P04	F8->F3,F4,F5,F6,F7
				Date	P21		
				Time	P22		
				City	P23		
				Centre	P24		
				Appointment_no	P25		
		Application Status()	F9	App_id	P8	NULL	F9->F8
				Processing	P26		
				Rejected	P27		
				approved	P28		
		Passport Status()	F10	Pending	P29	NULL	F10->F9
				dispatched	P30		

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.

The below algorithm provides a procedural approach for analyzing and evaluating the changes based on business policy enforcement

Algorithm violation detection (Change Specification cs, Service Logic L)

// Input: Change Request

// **Output**: Detecting policy violation

// Analyze the change specification cs for completeness and finiteness

for all cs !null

if (((rule | function | parameter) in cs is !complete) and ((rule | function | parameter) in cs is !computable)) **then**

Discard request

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

current_state:= previous_ state
break
else
// Map cs with the existing logic set L
if cs ∈ L **then**
Retrieve the corresponding rule, function and parameter, policies from L
for all P !null
if cs violates policy Pm **then** // Pm is the policy which the cs violates.
alert the analyst as "Policy Violation" // Ask for any modification in policy itself
if modify the policy Pm **then**
Pm := do the modification in Pm
P := P U Pm
else
discard changes
restore state
endif
else
Add cs to the L
L :=L ∪ cs
end if
end for
Retrieve the Dependency Set D and its associated Policies P
for all P !null
if cs violates policy Pm **then** // Pm is the policy which the cs violates.
alert the analyst as "Policy Violation" // Ask for any modification in policy itself
if modify the policy Pm **then**
Pm := do the modification in Pm
P := P U Pm
else
discard changes
restore state
endif
else
Add cs to the L
L :=L ∪ cs
end if
end for
else
Add cs to the L
L :=L ∪ cs
end if
end

Fig 1: Algorithm for Business Policy Enforcement

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

BEFORE IMPLEMENTING THE CHANGE REQUEST

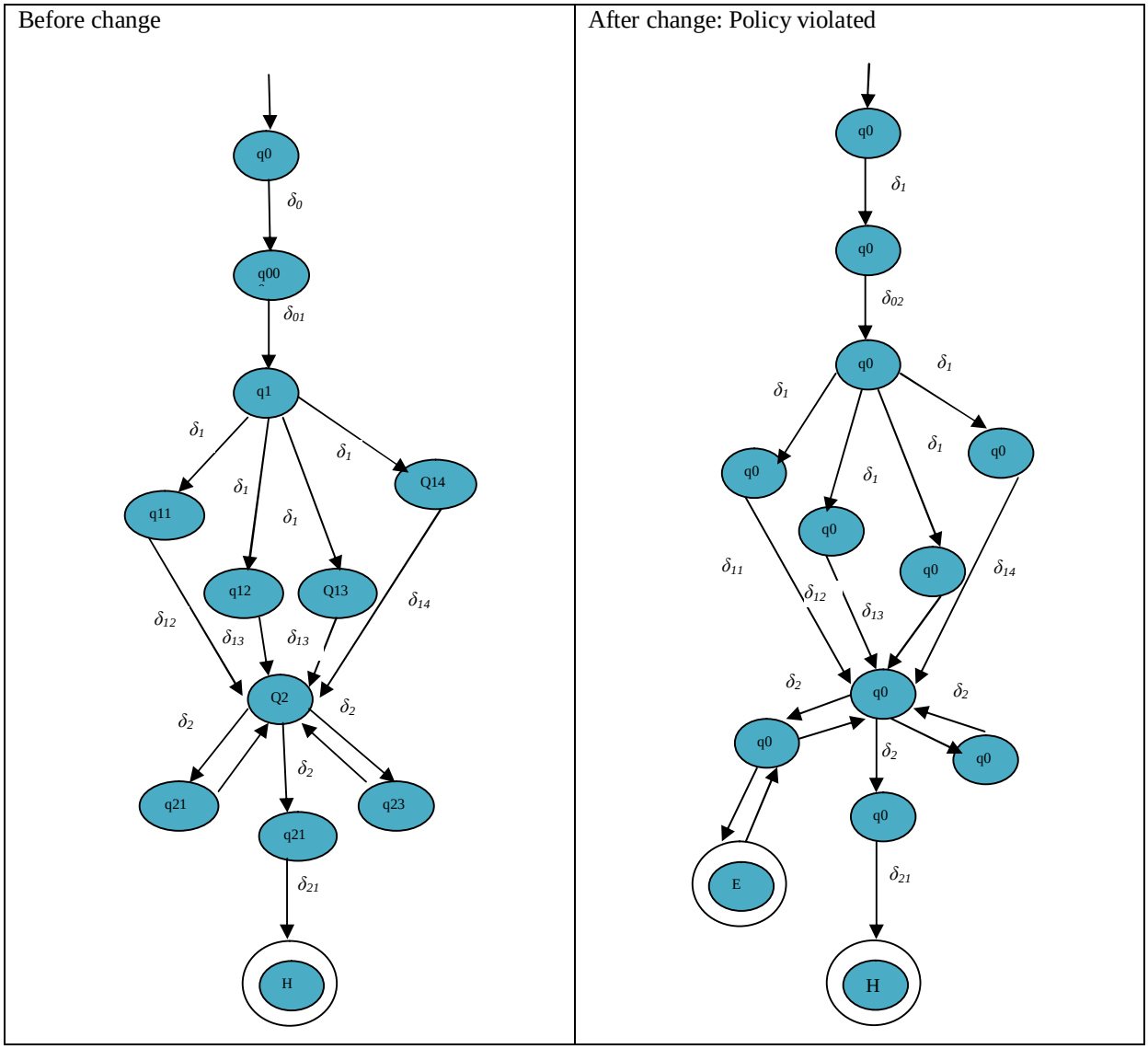

Fig 2: Finite State Machine before and after change

STATE TRANSITION TABLE BEFORE IMPLEMENTING THE CHANGE

Here the $\delta_0(q_{00}, \boldsymbol{\delta}_{01})$ represents the transition in which R2 represents the current state and δ_{01} represents the input for δ_0 and it is also the internal transition for δ_0. Within the internal transition $\delta_{01}(q_{00}, \{ P_8\}\{(q_{00}, (q_1 \}\{ PO1\})$, q_{00} represents current state and input includes three sets in which first set is for parameter and second is for dependency and third one is for policy.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Table 2: State Transition Table before implementing the change

Current State	Transition		Next State
q0	δ_0(R2, δ_{01})	$\delta_{01}(q_{00},\{P1, P_8\},\{ q_0, q_1\},\{ NULL\})$	q1
q1	δ_1(R3,{δ_{10}, δ_{11}, $\delta_{12,}$ δ_{13}})	$\delta_{10}(q_{11},\{ P_{8,} P_{14,} P_{15} \},\{ q_{00}\},\{ P05 \})$	q2
		$\delta_{11}(q_{12},\{ P_{16,} P_{17} \},\{ q_{00}\},\{ P01,P05\})$	q2
		$\delta_{12}(q_{13},\{ P_{18}, P_{19} \},\{ q_{00}\},\{ P05\})$	q2
		$\delta_{13}(q_{14},\{P_{20} \},\{ q_{00}\},\{ P02,P03 \})$	q2
q2	δ_2(R4,{$\delta_{20,}\delta_{01,}$ $\delta_{21,}$})	$\delta_{20}(q_{21},\{ P_{21}, P_{22}, P_{23,}P_{24}\},\{ q_{00}, q_{21}\},\{ P04\})$	q2
		$\delta_{01}(q_{23},\{ P_{26,} P_{27}, P_{28}\},\{ q_{00}, q_{21}\},\{ NULL\})$	q2
		$\delta_{21}(q_{22},\{ P_{29}, P_{30}\},\{ q_{00}, q_{21}\},\{ NULL\})$	H

AFTER IMPLEMENTING THE CHANGE

After implementing the change, the policy PO4 is getting violated. So control reaches the exception state which again rolls back to the prior state (i.e. state without implementing the change request).

STATE TRANSITION TABLE AFTER IMPLEMENTING THE CHANGE

Here the parameter P_{31} is inserted in the parameter set of the function F8 but due to policy violation, the control reaches the exception state E and rolls back to the prior state without implementing the change. Hence after the roll back, the P_{31} is eliminated from the parameter set.

Table 3: State Transition Table after implementing the change

Current State	Transition		Next State
q0	δ_0(R2, δ_{01})	$\delta_{01}(q_{00},\{P1, P_8\},\{ q_0, q_{01}\},\{ NULL\})$	q1
q1	δ_1(R3,{δ_{10}, δ_{11}, $\delta_{12,}$ $\delta_{13,}$ δ_{14}})	$\delta_{10}(q_{11},\{ P_{8,} P_{14,} P_{15} \},\{ q_{00}\},\{ P05 \})$	q2
		$\delta_{11}(q_{12},\{ P_{16,} P_{17} \},\{ q_{00}\},\{ P01,P05\})$	q2
		$\delta_{12}(q_{13},\{ P_{18}, P_{19} \},\{ q_{00}\},\{ P05\})$	q2
		$\delta_{13}(q_{14},\{P_{20} \},\{ q_{00}\},\{ P02,P03 \})$	q2
q2	δ_2(R4,{$\delta_{01,}\delta_{10}$})	$\delta_{20}(q_{21},\{ P_{8,} P_{21}, P_{22}, P_{23,}P_{24}\},\{ q_{00}, q_{21}\},\{ P04\})$	E

Thus the Business Policy Enforcement is evaluated through policy mapping. The percentage of policy violations detected is increased when FSM based evaluation is performed as shown in figure 3 (comparison has been done considering the same LCS and similar requests) which indicates that the evaluation process has been refined and has resulted in the increase in the detection of bugs.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

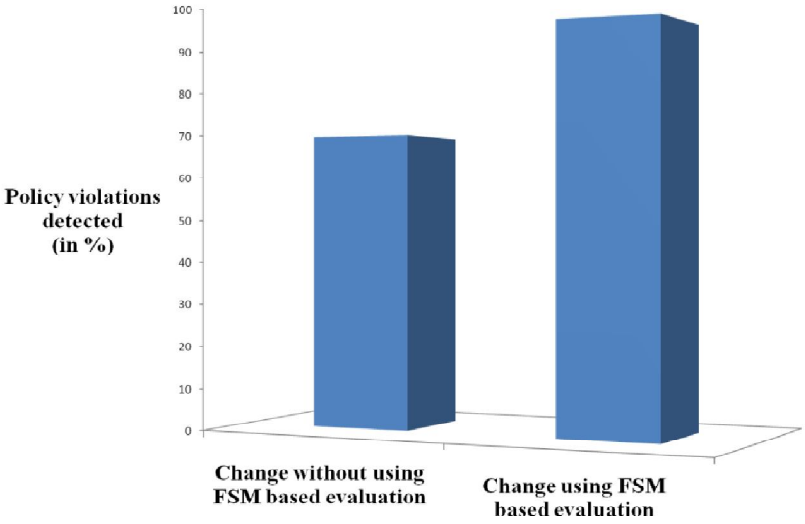

Fig 3: Percentage of policy violations observed

IV. CONCLUSION AND FUTURE WORK

The proposed methodology, Finite State Machine thus efficiently performs the evaluation of the changes made. Order of execution in LCS has been evaluated with the aid of the state transition table. The elucidation of the proposed approach using Passport system as the case study gives a clear idea of the change scenario and the evaluation of order of execution. The future enhancement is to include more factors for change evaluation which will aid in the assessment of the deviations in the functionality of an LCS after a change.

REFERENCES.

[1]. Salman Akram, Athman Bouguettaya, Xumin Liu, Armin Haller, Florian Rosenberg, Xiaobing Wu. A Change Management Framework for Service Oriented Enterprises, International Journal of Next-Generation Computing (IJNGC), Vol. 1, No. 1, pp. 1-077, 2010.

[2]. Xumin Liu, AthmanBouguettaya, Jemma Wu, and Li Zhou, "Ev-LCS: A System for the Evolution of Long-term Composed Services", IEEE Transactions on Services Computing, Issue: 99, ISSN: 1939-1374, 23 June 2011.

[3]. Salman Akram, Athman Bouguettaya, Xumin Liu, Armin Haller, Florian Rosenberg, Xiaobing Wu. A Change Management Framework for Service Oriented Enterprises, International Journal of Next-Generation Computing (IJNGC), Vol. 1, No. 1, pp. 1-077, 2010.

[4]. Minhong Wang, Huaiqing Wang, "From process logic to business logic—A cognitive approach to business process management", Elsevier journal of Information & Management, Vol. 43, Issue 2, pp. 179–193, March 2006.

[5]. Shuying Wang, Miriam A. M. Capretz. A Dependency Impact Analysis Model for Web Services Evolution, IEEE International Conference on Web Services, IEEE Computer Society, 2009.

[6]. Hua Xiao, Jin Guo and Ying Zou. Supporting Change Impact Analysis for Service Oriented Business Applications, IEEE International Workshop on Systems Development in SOA Environments (SDSOA'07), 2007.

[7]. H. Yahyaoui, L. Wang, A. Mourad, M. Almullah, Q.Z. Sheng,"Towards context-adaptable Web service policies", in Procedia Computer Science, Vol. 5, pp. 610–617, 2011.