

Web Service Business Policy Customization through Finite State Machine based Semantic Representation

Mekala.N

PG scholar

Department Of IT

Annai Mathammal Sheela

Engineering College Namakkal

Kavitha.K

Assistant Professor

Department Of IT

Annai Mathammal Sheela

Engineering College Namakkal

Ragupathi.V

Assistant Professor

Department Of IT

Annai Mathammal Sheela

Engineering College Namakkal

ABSTRACT

This paper proposes a finite state machine (FSM) based approach for Web service business policy customization that can address two key challenges: automated policy detection and evaluation and dynamic web service deployment. We have applied this technique on insurance domain in order to customize insurance policies for various types of customers. The business process for the insurance domain are identified and represented as finite state machine then the corresponding web ontology file is generated from the FSM. The use of web ontology engine at runtime enables dynamic deployment of a business process as a result of a business policy customization associated with the web services.

General Terms

Algorithms.

Keywords

Business Process Customization, Business Policy, Finite State Machine, Web Ontology Language (OWL).

1. INTRODUCTION

One of the burning issues in web service deployment is the customization of the business policies associated with the services. In the current scenario, there are no formal methods to facilitate automated detection of business policies and their evaluation. The proposed work guarantees this automation by the use of web ontology engine which ensures the dynamic deployment of the business process. The semantic information is made possible by the construction of the Finite State Machine (FSM) which eventuate the updation of OWL policy tags resulting in the detection and evaluation of business policies. Business Policy defines the scope or spheres within which decisions can be taken by the subordinates in an organization. The automated detection and evaluation of the business policy and the dynamic web service deployment has been made possible by the proposed system. The proposed technique has been elucidated in detail by applying it on the insurance domain in order to customize insurance policies for various types of customers.

2. RELATED WORKS

The works related to the focus of this paper are discussed in this section. Customization done during or after the runtime time, enabling the customers to customize the service and process during Requirement process, design process and testing the correctness of process logic while deploying process, modifying dynamically, substituting services and handling Runtime Exceptions according to the rules as services requested by customers has been proposed [1]. A

Change Management framework for making minor alterations to business logic but whose effect is more pronounced to the profitability of the organization has been proposed. The main aspect is that commercial entity is managed at business analyst's discretion and not at developers' discretion which saves time and cost [2]. A business process model is a model of one or more business processes, and defines the ways in which operations are carried out to accomplish the intended objectives of an organization. Business process modeling (BPM) in systems engineering and software engineering is the activity of representing processes of an enterprise, so that the current process may be analyzed and improved in future[3],[4]. BPM is typically performed by business analysts and managers who are seeking to improve process efficiency and quality. Business process modeling has always been a key aspect of business process reengineering (BPR) and continuous improvement approaches. [5]. Within Business Process Management all business objects, their attributes and their relations are defined in the Business-Object Model (BOM). The BOM is a result of semantic integration and is a representation of enterprise semantics. The content of the business model varies, depending on the type of business but the basic components are Infrastructure, Services, Customers and finance. A business process is a collection of related, structured activities that produce a service or product that meet the needs of a client. BPM is an approach to integrate a "change capability" to an organization - both human and technological. The activities which constitute business process management can be grouped into five categories: design, modeling, execution, monitoring, and optimization. The following people use the business object model: Business-process analyst, Business designer, System analyst, Designer. [6],[7]. Business Rule is a statement that defines or constrains some aspect of the business. This must be a term or fact, a constraint, or a derivation. It is 'atomic' in that it cannot be broken down or decomposed further into more detailed business rules. If reduced any further, there would be loss of important information about the business. [8], a number of leading-edge technologies have matured, which need to be incorporated into the logical solution architecture, such as portal, process choreography, business rule engine, enterprise service bus, Web services, service composition, etc. A new service-oriented model is proposed in this paper, to extend the traditional 3-tier architectural style and position the emerging technologies/products in the right places in the architecture structure. Web Service providers publish their Web service and invocation interfaces that they intend to offer by using Web Services Description Language (WSDL) and register the Web services to a common registration table located in

Universal Description, Discovery, and Integration(UDDI). Then, application programs discover the needed Web services and send the requests via invocation interfaces. After the response from a Web service provider, they invoke those services under Simple Object Access Protocol (SOAP) by using asynchronous messaging or Remote Procedure Call (RPC) mode. The changing user requests, demands the dynamic modification of the deployed web services at run time. This modification must ensure an uninterrupted service for the service consumers. This must be ensured for the fulfillment of the customers and to avoid the loss which will be incurred by the service providers if the service is interrupted. The user request may require configuration, modification, and change or source code management of the existing web service. [3][8] Business Process Modeling (BPM) technology continues to face challenges in coping with dynamic business environments where requirements and goals are constantly changing at the execution time. In practice, many organizations either focus their traceability efforts on functional requirements (FRs) or else fail entirely to implement an effective traceability process [9, 10]. Tracing Non-Functional Requirements (NFRs) has not been focused much. [15]Change management technique based on the integration between Object Oriented (OO) diagramming technique and Petri Net (PN) modeling language facilitates to increase the representation capability for OO modeling for the dynamic changes in the runtime instances. [9] The change management process should minimize the change completion time and maximize the change capacity. [4]Causes for changes include the introduction of new services or the alteration of goals. Dynamic reconfiguration is currently achieved in workflow specifications employing Web services using techniques that modify endpoint bindings and control structures. [5] A more challenging problem for Web Service reconfiguration is to reconfigure services dynamically on demand. A policy based framework supports three levels reconfiguration strategies that separate reconfiguration from application. This framework includes three conceptual layers: change instruction, reconfiguration management, and application management. The modification of Web services is a task that entails the modification and compilation of the source code, as well as the deployment of the new version of the Web service in the application server.

3. PROPOSED WORK

The figure 1 explains the proposed business policy customization technique. The Policy requirements are analyzed by the Business Policy Customization Manager which extracts the Web Service Source using the Service Locator. The Service Locator makes use of the Universal Description Discovery and Integration Registry and the Service repository comprising of the business process, business rules, business functions, business parameters and business policies. This is followed by the construction of Finite State Machine which eventuates the construction and

update of the customization tags and the OWL Policy tags. Business Policy Ontology depicts the business requirements and the Service Level Agreements followed by the Business Policy Evaluation and the Compilation and deployment of services.

OWL-S

OWL-S is a markup language used to describe the properties and capabilities of their web services in unambiguous, computer- interpretable form. Challenging issue in web services can be resolved using OWL-S like automated Web service discovery, execution, composition and interoperation. For this OWL-S provides means to semantically describe the prerequisites of a Web Service, the single steps it is composed of, and the way it is to be contacted. OWL-S provides semantics to web service in terms of three parts namely service profile, service process, service grounding. Service profile includes information about service provider, functionalities of service, and additional characteristics. Service provider's information contains contact details, agreements, payments etc., Functionality of service includes input and output, preconditions and their postconditions etc., and additional characteristics includes attributes like quality, performance, response time, constraints etc., in it.

Service process includes information about functionality of service in depth. It enlightens actual work of service in it. First we have to clear about the process types like atomic, simple, composite process. Atomic process can be able to act like individual service which can be invoked and executed on own. Simple process is also similar to atomic process but it is not invocable. It can be used in association with atomic or composite process. Composite process comprises of simple services together by hiding composition details. Similar to other two work flow languages, OWL-S also includes operations like sequence, split, split join, unordered, If-Then-Else, Iterate, Repeat-until, repeat- while.

Sequence operation is similar to previous workflow sequence, executed in lexical order. Splitconstruct are to be executed concurrently and it does not have any waiting state and synchronization. Split+Join is similar to split operation but it allows synchronization and wait for all activities to be completed. Unordered operation requires process enclosed with it to be executed in any order but all should be completed. If Then Else operation is similar to imperative programming language. Iterate operation is used in situation where it requires a process to be executed repeatedly. Service grounding includes information about how to access or invoke the atomic services. It contains details about communication protocols, message types, and other details like port information's which are necessary for contacting the service. Moreover it also contains abstract input and output of simple and atomic services and thus providing mapping between WSDL files and grounding detail. Any other description files can also be used for interface mapping.

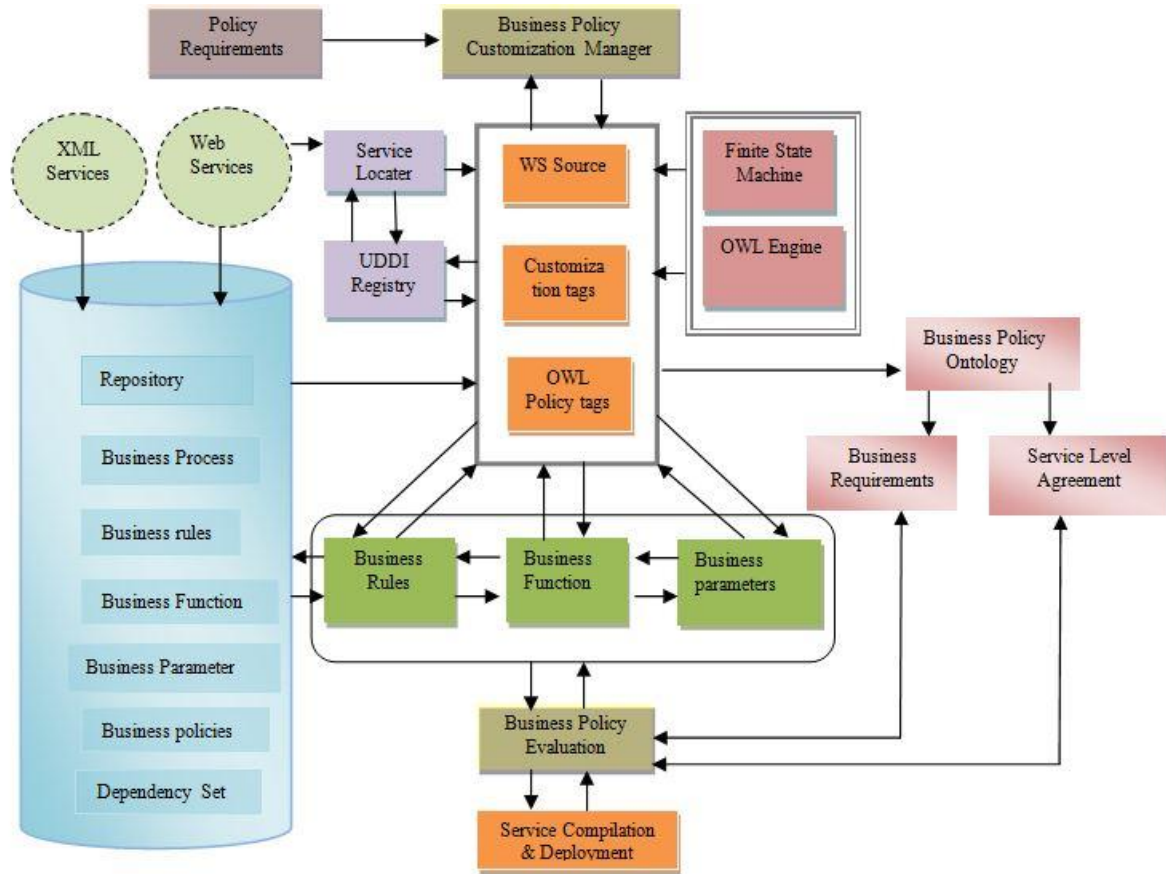


Fig 1: Proposed Business Policy Customization technique using Semantic Representation

POLICIES:

PO1 :PolicyHolder's Account should be created only if the age is valid between the limit of 18-50.

PO2: Sum assured should have minimum of 1,00,000 and these sum assured varies depending on the policy type chosen.

PO3: Initial amount paid during the start of policy should be fixed through the completion of policy. Any increase or decrease is not allowed.

PO4: In case of any sudden death within payment period ,full amount will be given to nominee.

PO5: If the payment not done correctly then the intimation is sent through address if no response then Policyholder's account is cancelled.

R1(CREATING POLICYHOLDER'S ACCOUNT)

F1- To create the PolicyHolder's account agent have to get the personal details such as their name, age, dob, resi_address, Qualification, working_address,annual_income,Nominee name, bnk_acnt_no/pin_no, Bank_name. P1 is the policyholder's name to whom the policy is scheduled. P2 is the age of the policyholder and it should be between the range of 18-50 because incase this range is not met then account should not be created. P3is the date of birth which is used to check whether their age is valid or not.P4 is the resi_address of policyholder to whom the information is sent. P5,P6 is the qualification and annual of policyholder, P7 is the Nominee name to whom the policy is given in case any financial

problem of their family. P9 and P10 are the details of bank_account_no and bank_name to which savings amount is deposited at the last date of maturity. There is no dependency for F1.

F2- After creating the policyholder's account we have to check the age whether it is valid or not. So F2 is dependent on F1.

R2 (SCHEDULING THE INSURANCE POLICY)

F3- used to add the policy we are going to schedule. P10 is unique id given to every holder. P11 is the Policy_Type which is the name of policy to start. P12 is the Policy_term_year and it varies depending on the policy type chosen and this policy term should start with 10 years. P13 is the mode_of_payment these parameter can be halfly, monthly, annually. P14 is the Sum assured and should have minimum of 1,00,000 and these sum assured varies depending on the policy type chosen. So the F3 is dependent on the F1,F2.

F4 - used to schedule the policy. P10 is unique id given to every holder. P11 is the Policy_Type which is the name of policy to start. P12 is the Policy_term_year and it varies depending on the policy type chosen and this policy term should start with 10 years. P13 is the mode_of_payment these parameter can be halfly, monthly, annually. P14 is the Sum assured and should have minimum of 1,00,000 and these sum assured varies depending on the policy type chosen. P15 is the start date of policy and P16 is the last date of payment. So the F4 is dependent on the F3.

F5- used to confirm the schedule of the policy. P10 is unique id given to every holder. P11 is the Policy_Type which is the name of policy to start. P12 is the Policy_term_year and it varies depending on the policy type chosen and this policy term should start with 10 years. P13 is the mode_of_payment these parameter can be halfly, monthly, annually. P14 is the Sum assured and should have minimum of 1,00,000 and these sum assured varies depending on the policy type chosen. P15 is the start date of policy and P16 is the last date of payment. P17 is the maturity date which is expiry date of policy. So the F5 is dependent on the F3.

R3 (AMOUNT SAVING)

F6- The amount paid for every year is checked using the parameter P10, P19, P20, P21. P10 is to get the information about the policyholder's schedule details. P19 is the initial amount that is paid and this amount should not be changed throughout the process. P20 is the number of payment made so far. P21 is the lastly amount paid. So F6 is dependent on F4 and F5.

F7- In case of any sudden death within payment period, full amount will be given to the nominee. P22 is used to get the when the risk have occurred. P23 and P24 is the date and

amount till which payment is done. So F7 should have to dependent on F6.

F8- Total amount to be paid up to the term year is saved and we used the parameters namely P10, P12, P14, P25, P26. P25 is totally how much amount we paid so far. P26 is the interest rate and this interest rate is increased based on the payment made. Interest calculation for month = amount earned * interest rate/12. Interest calculation for year = amount earned * interest rate/365.. So, F8 is dependent on F6 and F7.

R4 (BANK ACCOUNT)

F9- to deposit the whole amount we use the parameter namely P27. The bank details are stored already while creating the policyholder's details. F9 is dependent on F1 and F8 because inorder to get the fully payable amount with interest.

AFTER CHANGE

Suppose the change request is to Change the initial amount.

Orange arrow- what are the dependent function gets changed while doing the changes.

Violet arrow – po3 gets violated.

Fig 2: Before Change

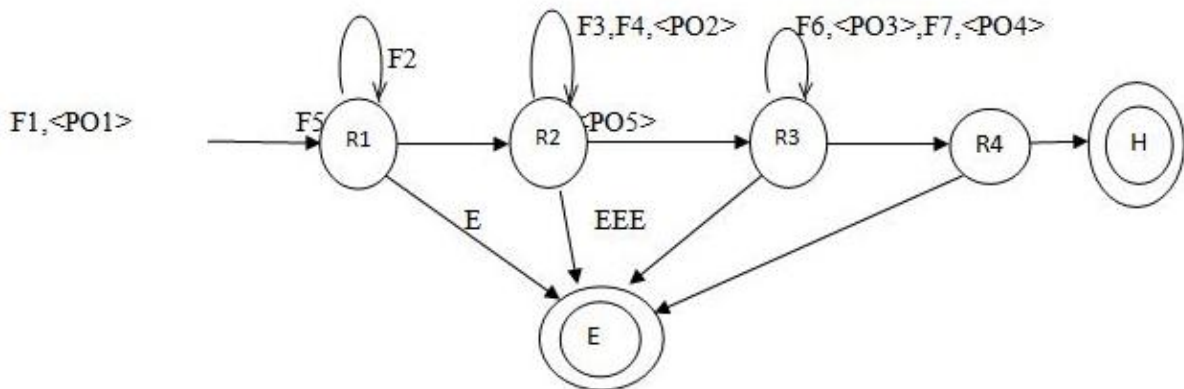


Fig 2: After Change

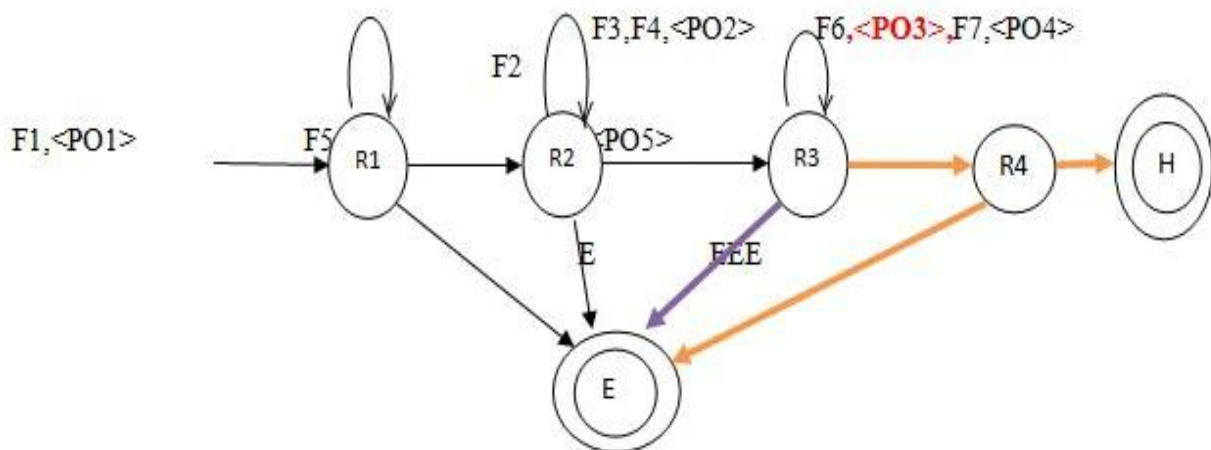


Table 1: Experimentation results

Rules		Functions		Parameters		Policy	Dependency
Policyholder's account creation		Create()	F1	Policyholder's name,	P1	Po1	NULL
				Age,	P2		
				Dob	P3		
				Resi_address,	P4		
				Qualification,	P5		
				Working_address	P6		
				Annual_income	P7		
				Nominee_name,	P8		
				bnk_accnt_no/pin_no Bank_name	P9		
		Valid_age()	F2	Age	P2	NULL	F2->F1
Schedule insurance policy	R2	Add()	F3	Policy_no,	P10	PO2	F3->F1,F2
				Policy_Type	P11		
				Policy_Term_year	P12		
				mode_of_payment	P13		
				Sum_assured	P14		
		schedule ()	F4	Policy_no,	P10	NULL	F4->F3
				Policy_Type	P11		
				Policy_Term_year	P12		
				Mode_of_payment	P13		
				Sum_assured	P14		
				Start_date	P15		
				Last_date	P16		
		Confirm_schedule ()	F5	Policy_no,	P10	NULL	F5->F3
				policy_type	P11		
				Policy_Term_year	P12		
				Mode_of_payment	P13		
				Sum_assured	P14		
				Start_date	P15		
				Last_date	P16		
Amount Saving	R3	Premium_paid_initially()	F6	Policy_no,	P10	PO3	F6->F4,F5
				Initial_amount	P19		
				No_of_payment	P20		
				Lastly_payable_amount	P21		
		Risk_occurs()	F7	Policy_no,	P10	Po4	F7->F6
				Risk_date	P22		
				Previously_payable_date	P23		
				Previously_payable_amount	P24		
		Total_amount_paid()	F8	Policy_no	P10	P05	F8->F6,F7
				Policy_Term_year	P12		
				Sum_assured	P14		
				Fully_payable_amount	P25		
Bank_account	R4	Total_amount_credited()	F9	bnk_accnt_no/pin_no Bank_name	P9	NULL	F9->F1,F8
				Full_amount_with_interest	P27		

4. CONCLUSION

This paper thus proposes a finite state machine (FSM) based approach for Web service business policy customization that can address two key challenges: automated policy detection and evaluation and dynamic web service deployment. The use of web ontology engine at runtime enables dynamic deployment of a business process as a result of a business policy customization associated with the web services. This results in better automation and enhanced customization of business policies in Web services.

5. REFERENCES

- [1] Thirumaran. M, Dhavachelvan. P and Shanmugapriya. R, Ontology based Dynamic Customization for Composite Web Services, *International Journal of Computer Applications*, 2013, DOI: 10.5120/11437-6577.
- [2] Thirumaran. M, Dhavachelvan. P, Asha.T and Lakshmi.P, Framework for managing Business logic of web services through Schema generation and Property evaluation, *International Journal of Computer Applications*, 2010, DOI: 10.5120/1317-1804.
- [3] Qing Yao, Jing Zhang, Haiyang Wang, Business Process-Oriented Software Architecture for Supporting Business Process Change. In *Proc. of International Symposium on Electronic Commerce and Security*, August 03-August 05, 2008.
- [4] Chengliang Wang, YaxinZhou,Juanjuan Chen, Extracting Prime Business Rules from large legacy system. In *Proc. of International Conference on Computer Science and Software Engineering*, 2008.
- [5] Fred A. Cummins. An Object Model for An Object Model for Business Applications Business Applications. In *Proc. of Electronic Data Systems Troy, Michigan*. Jan 2007.
- [6] Harry M. Sneed, KatalinErdos, Extracting Business Rules from Source Code. In *Proc. of4th International Workshop on Program Comprehension (WPC '96)*, 1996.
- [7] Rouvellou, I. Degenaro, L. Rasmus, K. Ehnebuske, D. McKee, B. Extending business objects with business rules. In *Proc. of Technology of Object-Oriented Languages, 2000.TOOLS 33.Proceedings. 33rd International Conference*, 5-8 June 2000.
- [8] Shan, T.C., Hua, W.W., “Solution Architecture for N-Tier Applications”,*In proc. Of IEEE International Conference on Services Computing, 2006.SCC '06*.
- [9] BassamAtiehRajabi, Sai Peck Lee, “Change Management in Business Process Modeling Survey”, *2009 International Conference on Information Management and Engineering*.
- [10] R. Baird M. Hepner R. Gamble M.T. Gamble “Reconfiguring Workflows of Web Services “ *Sixth International IEEE Conference on Commercial-off-the-Shelf(COTS) Based Software Systems(ICBSS'07)*.
- [11] Wenli Dong, “Dynamic Reconfiguration Method for Web Service Based on Policy”, *International Symposium on Electronic Commerce and Security, IEEE computer society*, 2008
- [12] Peng Cheng Xiong,Yu Shun Fan, and Meng Chu Zhou, “QoS – Aware Web Service Configuration” *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PARTA: SYSTEMS AND HUMANS, VOL.38, NO.4, JULY2008*
- [13] J. Cleland-Huang, “Toward Improved Traceability of Non-Functional requirements”, *Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering*, Long Beach, California, 2005, pp. 14 – 19.
- [14] Uttam Kumar Tripathi, Knut Hinkelmann, “Change Management in Semantic Business Processes Modeling”, *Eight International Symposium on Autonomous Decentralized Systems (ISADS'07)*.
- [15] Mira Kajko-Mattsson, Michal Tepczynski, “A Framework for the Evolution and Maintenance of Web Services “, *Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05)*.
- [16] Xiang Luo, KoushikKar, SambitSahu, PrashantPradhan, AneesShaikh, “On Improving Change Management Process for Enterprise IT Services” *IEEE International Conference on Services Computing*, 2008.