International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

Providing security for Web Service Composition using Finite State Machine

Raju R[1], Shanmugapriya S[2], Mahalakshmi P[3], Lalitha G[4]

[1]*Associate Professor and Head, IT Dept, Sri Manakula Vinayagar Engineering College, Pondicherry-605104, India*

[2, 3, 4] *Dept. Of Information Technology, Sri Manakula Vinayagar Engineering College, Pondicherry-605104, India*

[1]rajupdy@gmail.com

[2]priyasundar112@gmail.com

[3]mahazlakshmi154@gmail.com

[4]lalitha19ganga@gmail.com

ABSTRACT

The revolution impacted by Web Service as a solution to business and enterprise application integration throws light on the significance of security provided by Web Services during Web Service Composition. Satisfying the security requirements is truly a demanding task because of the dynamic and capricious nature of the Web. Web Service Composition associates web services to create high level business process that absolutely matches and conforms appropriately to the service requestor's needs. It involves customizing services often by locating, assimilating and deploying elementary services. Our paper proposes a policy based system for granting security during the process of web service composition. Policies defined for effective and secure composition analyze and verify the conditions under which the task of the web service is allowed or rejected. To achieve this specification, we make use of Finite State Machine model which clearly portrays the business and flow logic. Nodes in the Finite State Machine represent rules. Upon efficacious fulfillment of policies which are defined in the node access points, transition between rules is stimulated. A service composition is said to be successfully incorporated only if there is complete absence of policy violations when combining the policies of elementary services. The simulated FSM which extracts the rules and policies of the web services and correctly matches and satisfies the policy constraints defined in the access points ensure providing security for the composite web service.

Keywords— Composition System, Finite State Machine, Policy Manager, Web Service Composition, Quality Measurement Manager

INTRODUCTION

Service oriented architecture provides a reliable architectural framework which clearly depicts business coordination by planning business where participants collaborate in unison in a secure end-to-end business problem. In reality the most commonly followed and effective technology to implement SOA is evidenced as Web service. Web Services facilitate business applications efficiently available and accessible over the Web. They not only broaden the scope of business solution's accessibility but catalyses collaboration among multiple distributed applications via web service composition. Thus elementary services can be composed to provide the users with a single and effective application which exactly serves the requestor's needs. During Web Service composition, the services are orchestrated and delivered in an orderly manner according to the posted service requests.

A composite web service depends on the elementary web services for effective composition. Coordination among the elementary web services which may belong to varying root domains is a significant part to be considered. Service composition is adopted by almost all B2B applications across the Web which contains business specific workflows and/or orchestrated services. Web service composition has gained tremendous attention in recent years due to its flexibility to adapt quickly to changes in user requests and market conditions. Satisfying the security requirements during web service composition is a must in all applications over the web as the basic elementary services involved in composition may belong to varying domains. This is achieved by simulating the Finite State Machine Model.

Finite State Machine, actually an Artificial Intelligence technique has a mathematical root and hence used widely in matching patterns, sequential logic circuits and implementing computer programs. FSM can also alternatively be defined as a behavioral model comprising of a finite number of states, transitions between the states and actions similar to a action flow graph. Security for the composite web service can be attained by defining security policies for the basic elementary services which are to be composed. Consistency and rectitude among the policy specifications of the composed service as well as the elementary services should be considered as important. Simulating an FSM model for this purpose has a lot of benefits including clarity in specification of WSDL of elementary services and their associated sub services, Understandable depiction of flow logic and specification of security policies that were not defined previously in the WSDL description. Though combining services during composition based on user requests appears facile, it is not as easy to implement. Detailed composition architecture should be modeled. Also each service needs to be assessed to comply with its role as a composition member, and predicted service activities need to be studied in detail. Communication designs and paths, dealing with exceptions, transactions across services, business and security policies, and many more topics need to be effectively understood in order to make a composition exactly fulfill the service request.

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

Our paper focuses on achieving all the above stated terminologies through the policy based approach. The framework of our paper includes four components namely (i) Appeal Manager (ii) Assessment Manager (iii) Policy Manager (iv) Composition System(v) Flow Generator (vi) Quality Measurement Manager. Initially the services are created and published in the UDDI registry and are enlisted for use by the service requestor. After all the rules and policies' extraction the underlying FSM algorithm would be called for composing the requested web services. The policy manager is in charge of monitoring the access points where the policies are defined. Only when there is no policy violation the composition is said to be successfully achieved and the implemented FSM algorithm is said to be effective. The rest of the paper is organized as follows: Section 2 explains the motivation for our work dealing with previous and related work regarding Web Service composition and Finite State Machine implementation, Section 3 is about the Policy Based Approach, Section 4 gives the System architecture, Section 5 discusses the Results and Evaluation of our proposed system.

MOTIVATION

Our work is focused on securing Composite Web Service using policy based approach. We addressed that there is no clear rules in composing the policies of atomic and composite services without policy inconsistency. In the paper[1], they exiled rule for policy composition that can applied to any composite processes and afford security policy for Composition mechanism. They used two approaches: top-down and bottom-up for implementing the policy composition mechanism. In the top-down approach, the policies of composite services are considered without regarding the policies of elementary services. In adverse, bottom-up approach influences the necessary security requirements from the existing external services. BPEL definition is used as representation for composing Web Services and security policies are logically represented and they are transformed into prolog programs to draw inference security policy for the process. There is problem in give authorization for accessing the composite Web Service composed of Elementary Services. Hence, our work is concentrated on Access Control Policy (ACP) which means restriction to users to access the Web Services. XACML (eXtensible Access Control Markup Language) [4], and WS-policy are XML based representation. In these specifications, we need to add some extension to represent ACP for a Web Service since they are just framework for the representation of policy. BPEL definition for composite services are given as inputs to system and WSDL for service description for composite Web service and [2] XACML for Access Control Policy. They are transformed into predicates representing facts and composite services are drawn inference from those facts with the help of policy composition rule. But there occurs some inconsistency with policies of composite service and atomic services. The main drawback is that there is no automation rule in composing the policies of elementary services to obtain the policies for composite service. In the paper [3], they proposed a model for composite Web service to access its elementary services which belong to different security domain. They have composed the policies of composite web service without concerning the elementary services. To do this, we maintained, separate policy file for each elementary Web Service. To compose the policies of elementary services, only their policy files are taken into consideration. Each policy files have access control rules which contain condition element for restriction. If the condition value is true, the rule is satisfied and the user is allowed to access the service otherwise the access request is denied.

User's data should be protected when they are accessing the requested Web service. To secure user's data, the transport layer Security Protocol (SSL/TLS) [11].To secure the communication channel in terms to confidentiality, integrity and authentication, this layer is used. But in case of composite service, user's data entered in one elementary service is not protected after delivered to another Web service. The past histories of service invocation is used are used to make access control decisions [6]. It is often desirable to consider previous history of WS invocations when client attempts to access a web service composed of one or more elementary web services. Pure-past linear temporal logic (PPLTL) [6] which is declarative policy specification language uses Access Control Model.

In the paper [7], XSB prolog logic programming language represents the formal specification of security requirements and the corresponding assertions in exchanged messages. In the paper [8], to secure composite service, the specification of security requirements is integrated with the specification of composition of web service. The specification of interactions among the web services that participate in the composition according to various control flow patterns augmented with security related properties. In the paper [8], FSM model is used to analyze the reliability of the composed set of services by using the functional work flow process. Based on our research we have come to a conclusion that the use of security policy based approach for web service composition we can develop system more reliable and secured. Using FSM, we have several potential advantages such as reduced memory conception, faster and effective deployment of services, flexibility in searching and tracking of services, quicker response time and other key features which were either absent or minimal in the existing system.

POLICY BASED COMPOSITION AND FSM IMPLEMENTATION

The policy based web service composition for ensuring security during web service composition involves simulation of the Finite State Machine model which is a graph based data structure for structuring data based on an yes/no methodology. Simulation of the Finite State Machine models the rules of the services involved in composition as nodes in the Finite State Machine.

The policies are defined as check constraints at the access points of each node and are the transition conditions which foster transition from one rule (node) to another. Being in the initial input state (rule node 1 for example), the FSM model checks for transition conditions (policy check). If the transition conditions are met (all policies satisfied) it makes a transition to the next rule node as stated in the state flow diagram.

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

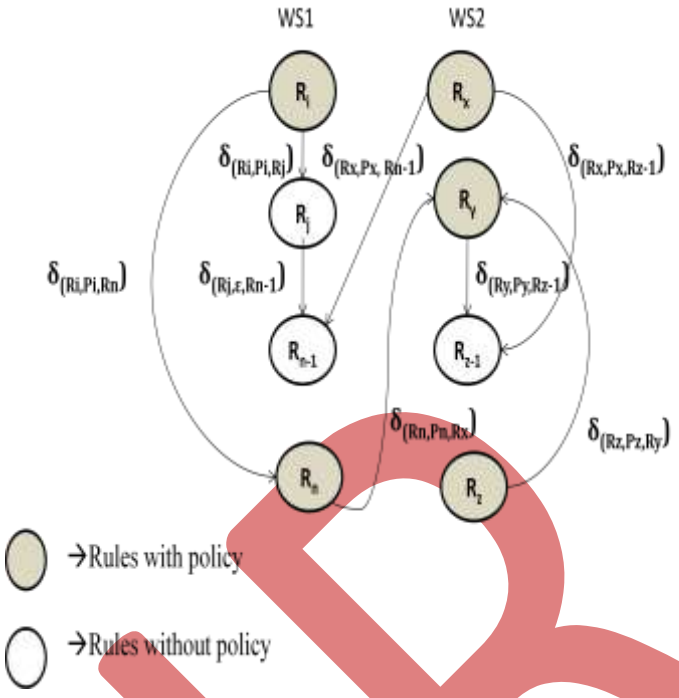


Fig 1. Finite State Machine depicting transitions between rules during the process of web service composition

This process echoes until the execution of the exit state after fulfillment of all the policies defined by the elementary services involved in composition. Thus Web service composition incorporates composite relationship between elementary web services.

Any service can call more additional services recursively to respond to a given user request. In addition any of the additional services can further call other services to finish the subtasks residing within the actual task. Thus each service that takes part in a composition plays the role of a service composition member.

To compose the web services and ensure security during the process of composition we may concentrate on two main areas: (i) FSM modeling which focus on how to correlate the elementary web services to take part effectively in service composition (ii). FSM technique implementation by executing the plan formulated during the FSM modeling period. When the service consumer requests a composite service, the policies defined for the elementary services are called in parallel provided the Service Level Agreement is satisfied. Composition and successful verification of policies in composite service is the key to achieve security in our system. The inputs to the system include the BPEL (Business Process Execution Language) process definition, WSDL (Web Service Description Language) descriptions and Access control Policies represented in XML (Extended Markup Language).

Consistency checks for the policies of the elementary and composite web service are also conducted in order to facilitate successful composition. The potential advantages of simulating FSM in our project are that it would provide the service requestors the flexibility in searching, tracking and consuming the web services in a secure way. Also it raises overall performance of the system by reducing the service halt time, service reply time and service consumption time.

Thus Finite State Machine is very critical in creating a secure composite service that calls the external services synchronously while servicing the requestor's needs. Business and security policies are defined for the atomic services. Rules are compulsory for every service while policies are discretionary within each rule. Correct prediction of the number of service composition factor with policy presence and without policy can facilitate the success of composition.

If web service composition is successfully achieved then evaluate performance based on significant performance metrics such as response time, availability, cost and security which concludes the fact that the service halt time, reply time and consumption time is at a satisfactory level. The state transition table showing the Finite State Machine's action flow sequence is given below.

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

TABLE 1 STATE TRANSITION TABLE OF SIMULATED FINITE STATE MACHINE

Point of Comparison	Web Services for composition	Source node	Input Condition	Transition notation	Destination node	Predicted Output
Existence of policy	(ser_1^ ser_2^…. ser_n)	R_x (x=1,2…n)	P_y (y=1,2…..m)	$\delta(R_x, P_y)$	R_z (z!=x) & (z=1,2…n)	a) Composition success; Policies satisfied b) Composition failure; Policy violation
Absence of policy	(ser_1^ ser_2^…. ser_n)	R_x (x=1,2…n)	ϵ	$\delta(R_x, \epsilon)$	R_z(z!=x) & (z=1,2…n)	a) Composition success; Policies satisfied b) Composition failure; Policy violation

The FSM model to achieve secure web service composition is based on deploying first service or demanded service from the Service storage repository and follow up steps to fulfill the requestor's objective. The umbrella activities encompass verifying the WSDL description if the input, output and mapping parameters of the web services to be composed are valid and authentic. Then reiteratively call the rules and associated policy xml definitions where they are clearly explained. Also verify policy consistency between basic elementary services and the composite Web Service.

Then validate using the Finite State Machine simulation. In addition determine the number of service composition factor with policy check and without policy check and then compose services based on this factor. If composition is found successful then evaluate quality parameters by invoking the graph based solution to predict performance and check whether service halt time, reply time and consumption time is low.

Verify Input, Output and Mapping parameters: $\{(In_i \ldots\ldots In_n),(Out_i \ldots\ldots Out_n),(Map_i \ldots..Map_n)\} \pounds (V_{par},F_{par})$

//where V_{par} is a set of valid parameters

//where F_{par} is a set of feasible parameters for composition

If validity is evidenced then:

Extract { $BRule_1 \ldots\ldots BRule_n$ }, { $BPol_1 \ldots\ldots BPol_n$ } from PReg

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

Perform policy matching: The check conditions need to be met for completing a successful transition. Rules established may or may not encompass policies.

SYSTEM ARCHITECTURE

The architecture framework of our proposal is depicted in Fig.1. The major components of our framework include (i) Appeal Manager (ii) Assessment Manager (iii) Policy Manager (iv) Composition System(v) Flow Generator and (vi) Quality Measurement Manager.

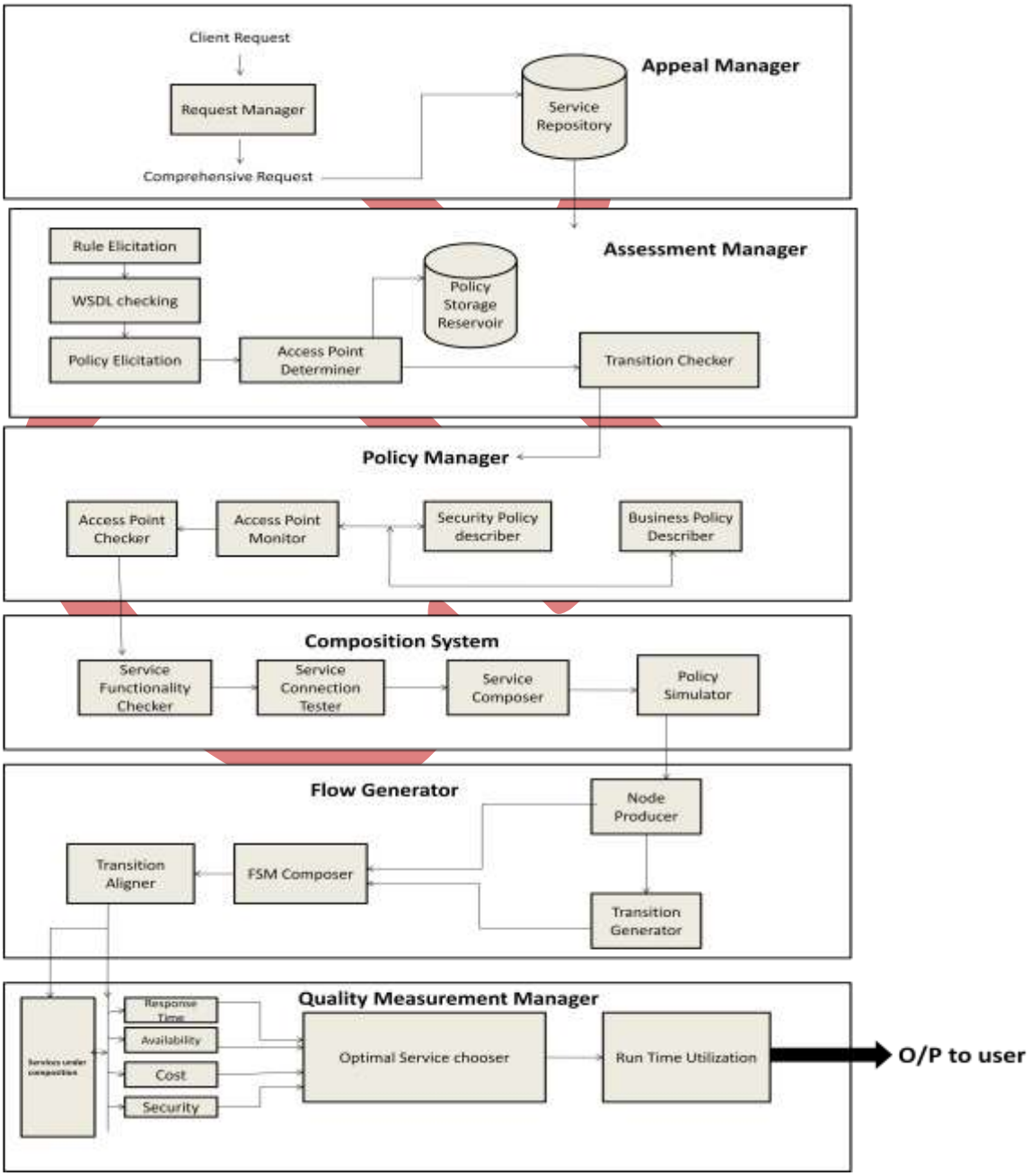


Fig 2. Architecture framework for secure web service composition using Finite State Machine simulation

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

In the Appeal Manager module, the user request is submitted to the request manager which processes the user request and generates a comprehensive message which it further submits to the service repository. The service repository is coupled with the assessment manager module where rules for the service are elicited, the WSDL of the web services to be composed are verified and corresponding policies are elicited and stored in the policy storage reservoir. The access points of the rule nodes are also determined.

The transition checker system verifies the correctness and reachability of the transition and calls the policy manager module where business and security policies are described and access points are monitored on a regular basis. The composition system is in charge of the web service composition process. Here the mission of the service is tested and the connection among the elementary services which are to be composed is checked for correctness. Also policy matching and fulfillment of policies estimates the success or failure of composition. The algorithm to implement the composition system is given below:

1. Start
2. /* VSer- set of valid services
3. ASer-set of authenticated services
4. FNA-Flag Not available
5. $SProv_n$ - Service provider [1….n]
6. No- Number of service providers
7. SerMan- Service Manager
8. PolRep- Policy Repository
9. SerCRep-Service Repository
10. PolSet- Policy Set */
11. Seek $SProv_n$ and Verify if $SProv_n$ €(VSer,ASer) then
12. // create, publish and deploy from registry
13. Call createWSer(WS_1….WS_n) and
14. pubService (WS_1….WS_n)
15. accessPtCheck()
16. For each service in SerCRep[] do
17. {
18. AccPoint ← find rule node access point of WS[]
19. If AccPoint match PolSet[] then
20. {
21. Compose from WS[]; // WS[] set of web services
22. }
23. Else
24. Set FNA[i]==1;
25. }
26. While PolRep!=null
27. Extract { $BsRul_1$…….$BsRul_n$ },
28. { $BsPol_1$…….$BsPol_n$ } from PolRep
29. //Check for presence of service
30. Fetch required web services from SerCRep
31. accessPtCheck();
32. If WS[N] not found then
33. Set FNA[i]=1; // for (i=0 to n)

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

34. Else

35. SetF NA[i]=0;

36. End

The next module of Flow generator contains the transition aligner whose functionality is to map transitions based on the input state and check conditions. The input to the transition aligner comes from the FSM composer. The FSM composer receives its processing data from the node producer and transition generator. The transition aligner also identifies the user requested services that are to be composed. The composition combinations are then predicted on the basis of cost, security, availability and response time which are the significant performance factors. Based on the result, the best composition is determined, selected and delivered to the end user.

RESULTS AND EVALUATION

The possible number of composed set of services is given by the formula:

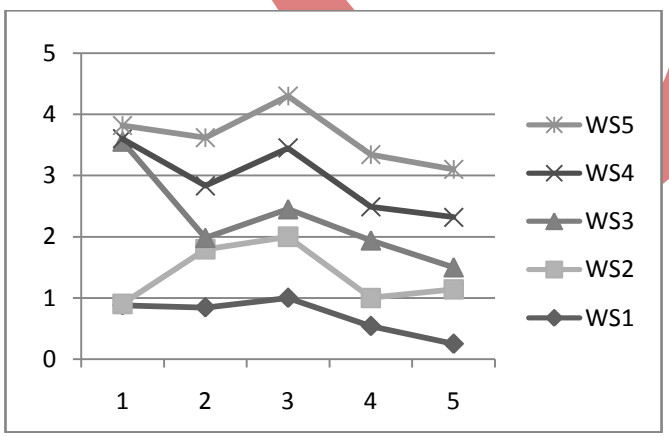
Composed service = = $WS_nC\ c_n$
WS_n = Number of Web Services (1…n)
c_n = Number of combinations (1….n)

The execution time texec(WServ, oprn) is calculated by the formula:

texec(WServ, oprn) = t(WSer, oprn) + 2 ¤ tl(WServ).

The cost of composed set of services is given by the formula:

CostC = α.KLOC^ β+φ



The ***availability*** of Composed Web Services is evaluated using the **Application Manager** tool.
Policy violation is used to evaluate the ***security*** factor. The policies are also composed while composing the number of Web services. Based on percentage of problem encountered in composing the policies of web services, security is evaluated.

Fig. 3. Performance Interpretation Graph

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

TABLE 2
PERFORMANCE INTERPRETER TO CHOOSE THE FINEST COMBINATION

S.No.	Web Services	Execution Time	Time (in percent)	Availability	Cost	Security
1	WS_1	0.88	0.845	1	0.54	0.25
2	WS_2	0.022	0.95	1	0.46	0.89
3	WS_3	2.654	0.187	0.45	0.94	0.36
4	WS_4	0.045	0.85	1	0.55	0.82
5	WS_5	0.216	0.786	0.85	0.85	0.78

TABLE 3
ESTIMATION OF SUCCESS AND FAILURE COMPOSITION USING FORMULA EVALUTION

No. of Web services User Requested	User ID	Web Service	Composed No. of Web Services	Successful Composition	Unsuccessful Composition
4	WS8526	4 –Web Services 2 - Combinations	$4C_2 = 6$	4	2
3	WS1234	3 – Web Services 2 – Combinations	$3C_2 = 3$	2	1
5	WS4365	5- Web Services 3 – Combinations	$5C_3 = 10$	5	5
6	WS5467	6 – Web Services 3 – Combinations	$6C_3 = 20$	15	5

CONCLUSION

This paper depicts the policy based approach for securing the composite Web Service. The definition for Composite service policy is given which invokes the elementary services. The consistency for the policies of composite service with elementary services is checked using Work flow graph of FSM model. We say secure Composite Web Service in terms of neglecting the policy violation in composing the web services. If the user provides data which violates the policies of composition and then it is shown to the user dynamically during request itself. The main advantage of our proposal that policies are checked dynamically while composing and depicted to the user by graphical form of representation such as Control Flow Graph and State transition table. Our proposal eliminates the drawbacks found in Petri nets such as high memory usage and more processing time. Implementation using First order logic which has the complexity in logical representation and abstract in actual logic is the drawback of our existing system. The performance factors such as security, cost, response time and availability are enhanced while implementing service composition policy using Finite State Machine.

Thus the graph based solution is depicted to show the performance of this type of policy composition rule. We have monitored the security policies on daily basis in our proposed system but there may be frequent change in the policies of web services. Further, we have an idea of updating the policies on regular basis continuously and to transform Composition Manager to Monitoring Manager so that security given for composite service will be more efficient and effective for users.

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

REFERENCES

[1] Fumiko Satoh and TakehiroTokuda "Security Policy Composition for Composite Web Services" IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 4, NO. 4, OCTOBER DECEMBER 2011.

[2] Thirumaran.M, Dhavachelvan.P, S.Abarna and Lakshmi.P, "Finite State machine based evaluation model for Web Services Reliability analysis" International Journal of Web & Semantic Technology (IJWesT) Vol.2, No.4, October 2011.

[3] Junqiang Zhu, Yu Zhou, and Weiqin Tong "Access Control on the Composition of Web Services" Proc. IEEE Int'l Conf. next generation of Web service practices, (NWeSP'06) 0-7695-2664-0/06 © 2006

[4] EXtensible Access Control Markup Language (XACML) Version2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml- 2.0-core-spec-os.pdf, 2011.

[5] HieuDinhVo, Dung Chi Phung, Vu Quang Dung,and Viet-Ha Nguyen "Securing Data in Composite Web Services" Proc. IEEE Int'l Conf. Knowledge and System Engineering, 978-0-7695-4760-2/12 © 2012 IEEE

[6] M. Srivatsa, A. Iyengar, T. Mikalsen, I. Rouvellou, and J. Yin, "An Access Control System for Web Service Compositions," Proc. IEEE Int'l Conf. Web Services (ICWS '07), pp. 1-8, 2007.

[7] C. Tziviskou and E.D. Nitto, "Logic-Based Management of Security in Web Services", Proc. IEEE Int'l Conf. Service Computing (SCC '07), pp. 228-235, 2007.

[8] A. Charfi and M. Mezini, "Using Aspects for Security Engineering of Web Service Compositions," Proc. IEEE Int'l Conf. Web Services (ICWS '05), pp. 59-66, 2005.

[10]Anca Muscholl and Igor Walukiewicz, "A lower bound on web services composition"LaBRI, Universit´e Bordeaux 1 and CNRS 351, Cours de la Lib´eration F-33 405, Talence cedex, France

[11] Rescorla, T.D.a.E. *The Transport Layer Security (TLS) Protocol*. 2008 July 2011]; Available from:http://www.ietf.org/rfc/rfc5246.txt.

[12] F. Satoh and T. Tokuda, "Security Policy Composition for Composite Services," Proc. Int'l Conf. Web Eng., pp. 86-92008.

[13] J.Y. Halpern and V. Weissman, "Using First-Order Logic to Reason about Policies," Proc. 16th IEEE Computer Security Foundations Workshop, pp. 187-201, 2003.

[14] D.D. He and J. Yang, "Security Policy Specification and Integration in Business Collaboration," Proc. IEEE Int'l Conf. Service Computing (SCC '07), pp. 20-27. 2007.

[15] Web Services Interoperability Organization (WS-I), http:// www.ws-i.org, 2011.

[16]Web Services Security: SOAP Message Security 1.1, http://www.oasisopen.org/committee/download.php/16790/ws-v1.1-spec-os-SOAPMessageSecurity.pdf,2011

[17] Thomas Erl, " Service Oriented Architecture Concepts, technology and design" ebook, pp.0-13-1858580. 2005.

[18] Matheus, A.: "How to declare access control policies for XML structured information objects". *Proceedings of the38th Hawaii International Conference on System Sciences*, IEEE (2005).

International Journal of Computers & Technology
Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

Author' biography with Photo

	Mr. R. Raju, M.Tech., is currently working as an Aossicate Professor in IT department of Sri Manakula Vinayagar Engineering College, Puducherry. He is research scholar, Bharathiyar University, Coimbatore.
	Ms. S. Shanmugapriya, pursuing B.Tech, final year in Information Technology department of Sri Manakula Vinayagar. Engineering College, Puducherry
	Ms. P. Mahalakshmi, pursuing B.Tech, final year in Information Technology department of Sri Manakula Vinayagar. Engineering College, Puducherry
	Ms. G.Lalitha, pursuing B.Tech, final year in Information Technology department of Sri Manakula Vinayagar. Engineering College, Puducherry