Finite State Machine Oriented Security Policy Composition for Composite Web Services

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*—Internet based applications nowadays focus majorly on how to compose web services with different root domains to exactly match the service requestor's need. Security in the access of the integrated composite service is an important non-functional requirement in order to achieve a successful and complete Business-to-Business application. In our paper we target on how to satisfy the security requirements for the composite web service by means of the policy based approach. Policies give the regulations that either approve or reject a behavioral deed. Thus there is a need of simulating a modeling technique or tool to achieve this specification. In this paper, we propose an FSM based behavioral model which clearly incorporates the business and flow logic. In actual we consider each elementary web service to be a node in the Finite State Machine. Security requirements are stated as security policies which are given as check constraints to access points of each node in FSM. Extraction of rules and policies of the composite web service and appropriate matching and satisfaction of the policies defined in the access points enable incorporating security for the integrated web process.**

***Keywords*—Finite State Machine, Policy Composition System, Composition manager framework, Web Service Composition, Service Level Agreement**

I. INTRODUCTION

Service oriented architecture structures business enabling a virtual federation of participants to collaborate in an end-to-end business process. Elementary services can be coordinated and composed to provide a single full fledged application to the users of the system. The services need to be orchestrated according to the requirements stated by the service requestor. Almost all B2B applications engaged over the web utilize the underlying concept of service composition. By statistics the most widely used and most effective technology to implement an SOA framework application has been Web service. Web service composition encapsulate business specific workflows or orchestrated services. Web service composition has been gaining increased attention in recent years due to its internal and external flexible capability to adapt swiftly to changes in customer requirements and market conditions.

There is an urging need to effectively satisfy the security requirements of a composite service. This can be implemented by simulating the behavioral model of FSM. Finite state machines, originally an Artificial Intelligence technique which is used to design computer programs and logic based digital systems. FSM can also be stated as a design model comprising of a finite number of states, transitions between the states and actions similar to a flow graph. Incorporating security for the composite web service can be facilitated by specification of security policies for the elementary services which are to be composed. The focus should be in maintaining consistency and integrity among the policy specifications of the composite service as well as the elementary services. Simulating an FSM model for this purpose has a lot of potential advantages including WSDL specifications of services along with their sub services, clear depiction of flow logic and definition of security parameters that were not specified earlier in the WSDL description. Also business incorporating service composition by simulating the FSM model attain reduced level of risks and process complexity and increased overall performance and outcome of a quality application.

The framework of our paper includes four components namely (i) Service Enlistment (ii) FSM Simulator (iii) Policy Checking (iv) Performance Evaluation. The services which are created and published in the registry are enlisted for deployment by the service requestor. After subsequent rules and policies' extraction the background FSM algorithm would be recursively called for composing the web services. The policy manager is in charge of defining the business and security policies. The policy manager also monitors and manages the access points where the security constraints are specified. Satisfaction of security policies and the overall performance evaluation portrays the efficacy of the implemented algorithm. The rest of the paper is organized as follows: Section 2 deals with the prior research related to Web Service composition and Finite State Machine implementation, Section 3 is about the Proposed System, Section 4 gives the System architecture, Section 5 explains the algorithm of FSM in web service composition, Section 6 explains the working model of our proposed system.

II. RELATED WORK

Our research can be divided into three sections namely composing elementary services, defining security policy for the composite service and FSM based solution. Our existing system is based on giving Security Policy for Composite Web Services [1]. In this paper [1], the composite service definition is written in BPEL and security policies are transformed into logical representation. The Access Control Policy is actually to give restriction for the user to access the Web Service. The ACP consists of set of operation name and the list of roles like (operation name, role list).

To represent ACP, we go for XML based specification like XACML [2], and WS-Policy. But both the specifications are just framework for the policy representation hence we need to add some extension to represent ACP for a Web Service. In paper [3] Separate policy files are created for each Elementary services which comprise to a single Composite Web Service. Each policy file consists of set of condition elements related to access control restrictions. If the value of condition element is true, rule is effective otherwise access request is rejected. This is actually for when user wants to access the composite web service without concerning the elementary services. Only the policy file of each elementary services are concerned while checking for access control. The main drawback is there is no automation for composing those policy files of all elementary services.

Roles are defined in Access Control policy [1] that can access an operation. Access Control Policies of composite service have consistency with atomic services. These policies are transformed automatically into predicates that represent them as facts. Through the Policy Composition Rule, policies for composite service are inferred. But the system finds some inconsistency among the policies of elementary services and composite service and actually the system does not support the automatic modification in the policies of composite service due to inconsistency and application semantics is the best way of modification.

Securing user’s data [4] for composite web service is taken into account which is to protect data sent between the services. This can be done by using transport layer Security Protocol (SSL/TLS) [10]. This layer makes the communication channel more secure in terms of authentication, confidentiality and integrity. But this layer cannot protect the data after being delivered to the receiver. The past histories of service invocation to make access control decisions are possible through declarative policy specification language uses pure past linear temporal logic [5] which is the standard Boolean predicates as well as support for temporal predicates. The formal specification of security requirements and the corresponding assertions in exchanged messages are built on XSB logic [6] which is based on Prolog programming language.

A framework is provided for securing BPEL specification using WS- Security and WS- Policy enhanced to SOAP message [7], integrating these specifications into specification of WS composition. FSM model [8] used to compose the services if policies of each elementary services are satisfied. This model is used to measure the reliability automatically whenever changes occur in the business logic.

III. RESEARCH PROPOSAL

Finite State Machine model is a graph based data structure which depicts the decision model algorithm which is purely based on an yes/no principle. If the input action conditions are met it executes the input action and checks for transition conditions. If in turn, the transition conditions are fulfilled the node makes a transition to the specified state. This process repeats until the execution of the exit state after satisfaction of the exit action conditions. Web service composition establishes composite relationship between a collection of web services. Any service can enlist one or more additional services to complete a given task request. Further any of the enlisted services can call other services to complete the subtasks within the actual task. Therefore each service that participates in a composition assumes an individual role of service composition member.

To compose the requested web services and incorporate security we may simulate the FSM behavioral model which focuses on two main aspects: 1. FSM design which focuses on synthesizing a specification of how to coordinate the elementary web services to fulfill the client request and satisfaction of the defined security rules and policies 2. FSM implementation by executing the specification produced by the FSM design. When the service consumer proposes a request for service invocation the business and security policies are called in parallel provided the Service Level Agreement is satisfied by the consumer.

The next step involves composing and validating policies in composite service by implementing a policy composition system. The inputs to the policy composition system are the process definition stated in BPEL(Business Process Execution Language), Service descriptions in WSDL(Web Service Description Language) and Access control Policies represented by XML(Extended Markup Language). The system then checks for any inconsistencies in the policies of the elementary and composite web service. After checking the absence of any policy inconsistencies, the composite service would be available for consumer deployment. The advantage of using FSM in our project is that it would provide the users the flexibility in searching, tracking and retrieving the web services in a secure and reliable manner. Also it enhances overall performance of the system by reducing the service interruption time, service response time and service consume time.

IV. SYSTEM ARCHITECTURE

The composition manager framework as shown in Fig.1 is described elaborately as follows. Initially the service provider creates the service and publishes it in the central registry for deployment by the service consumer.

In the service assessment module of the composition manager framework, the rules or conditions to be satisfied for the concerned web service are extracted in an appropriate data representation understandable by the users, then the business and security policies are extracted which clearly defines the scope and spheres within which the service can be deployed by the user.

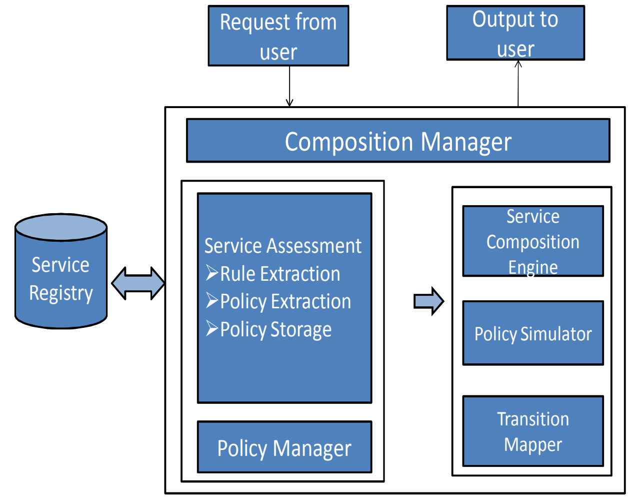

Fig.1 The composition manager framework

The service composition engine composes all the elementary web services and calls the graph based solution of FSM. The policy simulator and verifier module simulates the FSM model to implement secure web service composition. The transition mapper directs the transition from one state to another (one web service composed with another) in the FSM model.

After successful composition of web services and satisfaction of policies by the composition manager framework, the service consumer would be able to effectively deploy the service requested with consummation of the Service Level Agreement.

V. FSM IN WEB SERVICE COMPOSITION

Finite state Machine is very valuable in implementing a secure composite service that invokes the external services in parallel and symmetrically in the process of fulfilling the requestor's need. The security policy for a composite service consists of a composite Access Control Policy for the atomic services. The algorithm for FSM which implements the composition manager framework depicted in the previous section is shown below:

```
Begin
// V- set of valid services
// A-set of authenticated services
// NA-Flag- not available
// $SP_n$ - Service provider[1….n]
// N- Number of service providers
// SM- Service Manager
// PSR- Policy Storage Repository
// SCR-Service Central Repository
// PS- Policy Set
Determine $SP_n$ and Check if $SP_n$ €(V,A) then
Call create ($WS_1$….$WS_n$) and publish ($WS_1$….$WS_n$) // create, publish and deploy from registry
PolicyCheck()
For each service in SCR[] do
{
AP ← find policy access point of s[]
If AP match PS[] then
{
Compose from s[]; // s[]    services
}
Else
Set NA[i]==1;
}
While PSR!=null
Extract { $BR_1$…….$BR_n$ }, {$BP_1$…….$BP_n$ } from PSR
Fetch required web services from SCR //Check for availability
PolicyCheck();
If s[N] not found then
Set NA[i]=1; // for (i=0 to n)
Else
Set NA[i]=0;
End
```

VI. WORKING MODEL OF FSM

The FSM behavioral model to achieve secure web service composition is based on retrieving the first Web service or requested Web Service from the Service (composition) registry and steps adopted to fulfill the requestor's needs. The process includes checking the

WSDL service description file if the input, output and mapping parameters of the services enlisted for composition are valid. If there is validity or feasibility then invoke the rule and policy folders where the business and security rules and policies are defined. Check if there is a policy consistency between elementary services and composite Web Service. Then verify using the Finite State Machine simulation algorithm. Estimate the number of service composition factor with policy enforcement and without policy and then compose services based on this factor. If composition is successful then go for checking performance by calling the graph based solution to evaluate performance and check whether service Interruption time, response time and consumption time is low.

Check Input, Output and Mapping parameters:

$\{(I_i\ldots\ldots I_n),(J_i\ldots\ldots J_n),(K_i\ldots\ldots K_n)\}$ £ (V,F)

//where V is a set of valid parameters

//where F is a set of feasible parameters for composition

If validity is approved then:

Extract $\{BR_1\ldots\ldots BR_n\}$, $\{BP_1\ldots\ldots BP_n\}$ from PSR

Check for policy matching:

Illustration of a Web service which is considered as a node in Finite State Model and checking for policy match to make a transition from start state to final state.

Rule2:1(f)

$$\text{Rule1: o(s)}\quad \begin{array}{c|ccc|} & P_i & P_j & P_k \\ P_i & 1 & 0 & 0 \\ P_j & 1 & 1 & 0 \\ P_k & 1 & 1 & 1 \end{array}$$

Also the check constraints for transition need to be met for committing a successful transition. Rules defined for the services may or may not contain policies.

Estimate : The number of service composition factor with policy enforcement and without policy are identified and then compose services based on this factor

TABLE I
FSM STATES, TRANSITIONS AND ACCESS POINT CONDITIONS

	Input State	Check conditions	Output State

Web Services with policy enforcement (WS1,WS2.. WSn)	*Rule{***1..n***}*	*Policy{1….n}*
		*TransitionCondition{*1..n*}*
			*Rule{*2..n*}*

Web Services without policy enforcement (WS1, WS2.. WSn)	*Rule{*1..n*}*	*TransitionCondition{1..n}*	*Rule{*2..n*}*

Finally evaluate performance by calling the graph based solution and ensure the efficacy of the implemented algorithm.

VII. CONCLUSION

This paper mainly focus on providing security for composite web service by using the policy based approach. There is a policy file for each elementary service and while composing the services, the policies of all elementary services are also composed. When the requestor requests for composite service, the policies of composite service have to be satisfied in order to consume that service. In the existing system implementation using First Order Logic there are several backlogs such as complexity in the representation of logic and abstraction of actual logic.

Using FSM model for composition enhances performance and the composition logic can be easily incorporated and FSM model can be generated for the logic. A graph based solution is also proposed to evaluate performance of composite service by checking whether service interruption time, response time and service consumption time is low.

VIII. FUTURE WORK

Our algorithm describes how the services are composed with ultimate efficiency and time preservation. In our proposed system the security policies are monitored on daily basis and updated on daily basis. But Security policies change randomly throughout the day. Frequent monitoring of policies is important so that the services can be provided still efficiently and more securely. We have an idea of monitoring the policy details on regular basis continuously and update it and also to convert the composition manager to monitoring manager and provide still efficient services to the users.

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] 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

[3] 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.

[4] 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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9]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

[10] Rescorla, T.D.a.E. *The Transport Layer Security (TLS) Protocol*. 2008 July 2011]; Available from:http://www.ietf.org/rfc/rfc5246.txt.
[11] F. Satoh and T. Tokuda, "Security Policy Composition for Composite Services," Proc. Int'l Conf. Web Eng., pp. 86-92008.

[12] 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.

[13] 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.

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

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

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