Context and Trust Aware Workflow Oriented Access Framework

[1]Tapalina Bhattasali, [2]Nabendu Chaki, [3]Rituparna Chaki
[1,2]Department of Computer Science and Engineering
[3]A. K. Chaudhury School of IT
University of Calcutta, Kolkata, India
[1]tapolinab@gmail.com, [2]nabendu@ieee.org,
[3]rchaki@ieee.org

Khalid Saeed
Faculty of Computer Science
Bialystok University of Technology
Bialystok, Poland
khalids@wp.pl

***Abstract*— Service oriented computing (SoC) changes the way of conducting business as these services are often available on a network. As traditional access control approach may not work in the changed environment, protecting business resource from misuse is a big challenge. Again, static allocation of access right to users will not be an efficient solution as SoC environment changes with time. This paper focuses on design of dynamic access control approach for business process. Here, we propose a context and trust aware workflow oriented access framework. Proposed approach focuses on inter-component relationship where phases are executed either in online or offline mode to avoid performance bottleneck. The concept of static binding in role based access model is extended to support dynamic access control by including context awareness and trust relationship between owner and user. Trust is either directly or indirectly dependent on service level agreement (SLA) compliance, quality of service, reputation and provenance (historical data). In this paper, the framework is designed by mapping proposed access model to workflow instances at run-time. It is validated by workflow net model, where workflow instance can be successfully executed without any interruption and can satisfy soundness property while incorporating proposed access control approach.**

Keywords- dynamic access; workflow; context; SLA; trust; quality of service; service oriented computing

I. INTRODUCTION

"Any" paradigm supported by ubiquitous computing allows anyone to access objects at any time and from anywhere. This type of access should be carefully controlled to meet security and privacy requirements in dynamically changing service oriented computing environment. One of the key challenges in this type of environment is to design of effective access control approach. Designed model needs to protect resource and service against unauthorized access according to a security policy. It verifies whether a subject to object binding is allowed to carry out or not at run time based on several application specific criterions. In SoC, workflow is used to realize an automated business process by representing execution of multiple relevant tasks in an organized way to meet business objectives. However, growing complexities of business processes demand more flexibility in workflows. Flexibility can be achieved by adopting changes around the environment without changing its originality. This concept can be implemented by considering a framework, which can realize dynamic mapping between authorized users, roles, services and permissions during execution of workflow instances.

This work aims to maintain flexibility as well as security in service oriented computing applications, where anything is considered as a standard service. Services are invoked by huge number of subjects having temporary role and authorization procedures need to be passed through several security domains. Here static binding between subject and object is changed to dynamic binding by means of context and trust aware workflows. Workflow instances are defined and validated by Petri nets. Coverability property of resulting workflow nets are analyzed to verify successful execution of workflow instance after incorporating proposed access control approach, which in turn also analyzes compliance with the service level agreement. Soundness in workflow net model implies that all the tasks are successfully executed in an order, while considering dynamic mapping of access control approach and there are no major flaws in the specification of proposed access framework to interrupt the basic flow of tasks. Valid users can access service at right time under several constraints to realize that flexible operations are performed and flow of tasks goes on. As a result, execution of the application-specific workflow gets more secured, as probabilities for unauthorized data access and data misuses are reduced.

The major contribution of this paper is to propose a theoretical approach that integrates inter-dependent information either in off-line or in online mode to validate successful execution of workflow instance in service oriented framework. The rest of the paper is organized as follows. Section II presents few related works on this domain. Section III identifies scope of the work. Section IV introduces proposed access framework in service oriented computing environment. In section V, the proposed framework is defined as a workflow net model and it is validated using WoPeD simulator. The paper ends with concluding remarks in section VI.

II. RELATED WORKS

Research on access control approach designed for heterogeneous service-oriented computing environment is becoming a very interesting topic in the field of security. There are several access control approaches; each of which has its own merits and demerits. Due to page limitation few popular approaches are mentioned here. Most commonly used access model are - role based access model (RBAC) [1] and attribute based access (ABAC) model [1]. However major drawback of these types of access models is objects are assigned to subjects in a static manner that cannot be changed

DOI reference number: 10.18293/SEKE2016-179

with the change of environment. Assigning access rights dynamically can be achieved by context awareness. Therefore, concept of RBAC and ABAC are extended [2, 3, 4] to context aware RBAC and context aware ABAC respectively. In spatial RBAC [5], access to service is granted based on the location and assigned roles of users. In rule based RBAC (RB-RBAC) [6], users are dynamically assigned based on security policy. In situation aware RBAC (SA-RBAC) [7] users are granted to roles and roles are granted to permissions based on the situation information. In trust aware access control mechanism [8] for IoT (TACIoT), lightweight authorization mechanism and a novel trust model are specially devised for IoT environment. In [9], focus is given on synchronization of the workflow and the authorization flow. This is done by assigning time slots to the tasks in the workflow. After analyzing few recent works on this domain, it can be said that there still exist a research gap to balance between flexibility and security.

III. SCOPE OF WORK

Motivated by several identified research challenges [10], major problem can be defined as: How to dynamically bind subject to object according to inter-dependent information considered for a workflow instance that satisfies soundness property of workflow net? In order to solve this issue, a context and trust aware workflow oriented access control approach is considered here as scope of work. It includes design of a dynamic access framework for service oriented computing-

- To bind access right to a subject and object pair for a workflow instance according to change in context and evaluated trust value depending on service level agreement and quality of service.
- To validate access framework with workflow net simulator to ensure that workflow instance goes on smoothly and consistency is preserved in access control approach, if and only if it can satisfy workflow net properties (like well-structured, sound, live, bounded, and deadlock-free).

IV. PROPOSED APPROACH

This section introduces proposed context and trust aware workflow oriented access framework that depends on several components. Proposed approach is novel in the sense that context-aware dynamic access approach based on inter-component relationship maps with workflow instance and is defined by simple workflow net model. Thus, it can be easily implemented for a real life scenario. Besides, it can capture status of workflow instance for deriving successful execution of proposed access model to ensure safety and validation of the approach. It includes design of a dynamic access framework based on context information and evaluated trust value on workflow management.

A. Context and Trust Aware Access Approach

It integrates the concept of workflow based access control, context aware role based access control, trust based access approach control, and SLA aware trust model. Proposed access approach can be defined as a seven phase and nine tuple access model. Among the seven phases, phase 1 is only considered for off-line mode processing to avoid performance bottleneck. This approach is designed to meet the basic requirements of access control in service oriented computing framework. It supports fine granularity, scalability, context-awareness and capable to adapt run time changes. Figure 1 represents proposed access control approach.

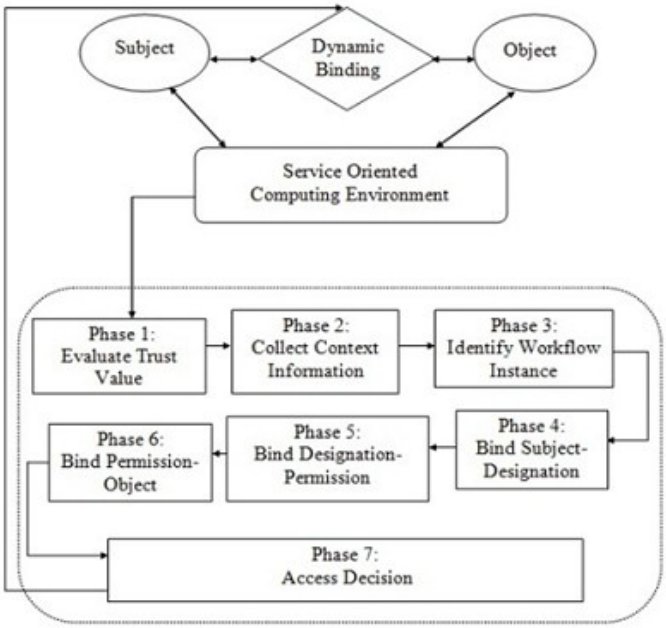

Figure 1. Context and Trust Aware Access Control Approach

Figure 2 represents basic building block of proposed access model.

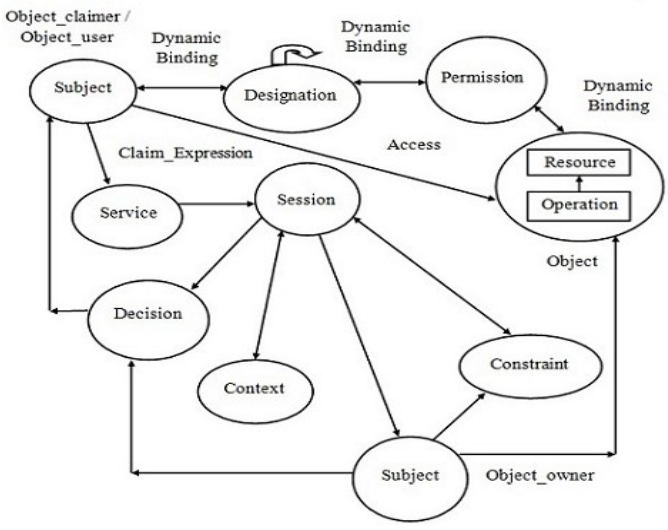

Figure 2. Building Block of Proposed Access Model

Proposed Access Model ={S, O, D, Per, Oper, Auth, Po, E, WF(t)}, where S → subject, O → object, D→ designation, Per→ permission, GL → granularity level, Auth→ authorization decision, Po → policy, Oper → operations, E→ expression, WF(t) → workflow instance. The major elements of proposed access model are discussed below.

S: Entity that takes action on an object. It is categorized into object_owner, object_claimer, object_user, relevant_entity.

O: Entity that is acted upon by a subject. It can be either resource or service. They are protected by set of policies.

D: It represents subject's functional role at a specific instance. There are several classes of D. Intra-class D can be easily interchanged for a workflow instance and interchange of inter-class D is based on dependency between two classes. Exchange of D within intra-class and inter-class is defined by designation partition rule, where two designations cannot be assigned to same subject within a session of workflow instance.

Per: This is the approval to perform certain operations on objects by the subjects. This is generated by access-right generator module. Per set includes {read, write, execute, modify, activate, deactivate, allow, disallow, upload, download}.

Oper: Per and Oper are related to each other. Based on Per, Oper is executed on object by subject at a time.

Auth: Decision of authorization is generated to enable dynamic binding at run time.

WF(t): It represents executable workflow vector at session time t with the lifetime T_1 to specify workflow instance.

Context(c): It is related to each entity involved in a workflow instance of the application. Context generally depends on {Time, Location, Environmental parameters,…} during its execution time.

Constraint (cons): It is expressed as a regular expression cons: = union (clause$_1$, ……, clause$_n$) ,
where clause:= intersect(cond$_1$,….,cond$_m$), and cond:= <c> <op> <context_val> AND <trust_val>, where op is a logical operator from set { <, >, <=, >=, ≠}.

Po: Set of rules is applied to model to get service as expected. It can be represented as a pentuple. Po: = < S, D, O, Oper, cons>.

E: These are used to express either claim_request or inter-dependent information in order to specify binding between subject-designation, designation-permission and permission-object.

Object_Access (OA): It can be represented as a quadruple. OA:= <S, Oper, O, dc>, where dc implies dynamic context considering every context within context set. OA is granted if and only if all tuples of it evaluate true under dc.

Hierarchical_Designation (HD): Designation is considered in a hierarchical manner to apply the concept of inheritance. If permission is assigned to a lower class of designation, then it is also assigned to all the higher class designations.

Object_Hierarchy (OH): It supports a subject to access different granularity levels of objects in a hierarchical manner. If a subject has the right to access object with the highest granularity level, then it has also right to access the lower granularity levels of that subject.

Dynamic_Binding (DB): It is a many to many mapping between subject and object at run time. It is mainly of three types- subject to designation mapping, designation to permission mapping and permission to object mapping.

Access Dependency (AD): It represents the relation between access decisions in workflow instances. Such as,

- AD2 can be activated only after AD1 is finished.
- AD1 and AD2 must be mutually exclusive for a workflow instance.
- Permissions granted to AD1 can be delegated to AD2, when AD1 is aborted.
- AD1 and AD2 must be granted to two different subjects.
- AD2 can be activated only after AD1 is defeated.

Figure 3 represents inter-dependency among multiple components of proposed approach.

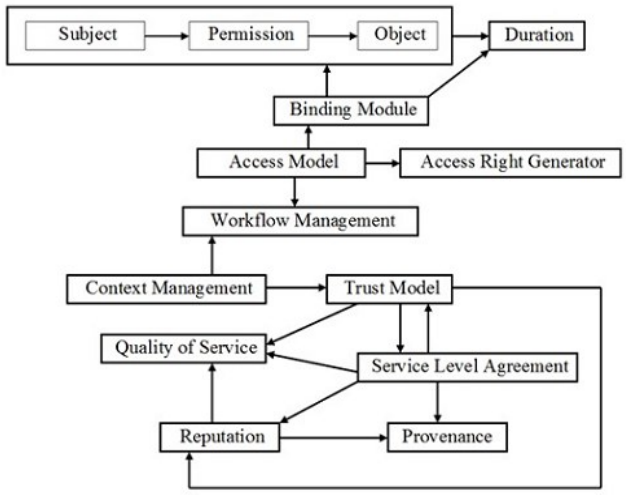

Figure 3. Inter-Component Relationship of Access

Inter-component relationship of access model is identified below. Left hand sides represent determinants and right hand sides represent dependents.

Provenance → {Reputation, Service Level Agreement}
Reputation → {Service Level Agreement, Trust Model}
Service Level Agreement → Trust Model
Quality of Service → {Reputation, Service Level Agreement, Trust Model}
Trust Model → {Service Level Agreement, Context Management}
Workflow Management → {Context Management, Access Model}
Access Right Generator → Access Model
Binding Module (operation, expression, designation) → Access Model
{Subject, Permission, Object} → Binding Module (operation, expression, designation)
Duration → {Binding Module, Subject, Permission, Object}

B. Integration Logic based on Inter-Component Relationship

Integration logic of inter-component relationship works in two modes- off-line and online mode to avoid performance bottle-neck that may arise if components are integrated during dynamic mapping at online mode. Procedural logic of proposed access approach is presented below for both off-line and on-line mode.

Off-line mode:

begin

1. construct SLA between subject S_m and S_n
2. set trust_set:= {1:1 trust, third_party trust, derived trust}
3. select trust_type
4. construct soft trust vector for each subject pair // soft trust is trust without hard copy proof
5. initially set provenance to NULL
6. check QoS parameters
7. compute reputation from (provenance, QoS) value
8. check SLA_compliance (provenance, reputation, QoS, trust)
9. evaluate trust (availability, degree of separation, regulatory compliance, recovery probabilty, successful access)
10. construct SLA aware trust
11. evaluate inter_component trust (SLA_compliance, reputation, QoS) for trust_set
12. trust_val= union (trust , inter_component trust)
13. store trust_val //to be used in online mode
14. update provenance value, reputation value, trust_val
15. evaluate trusted_relation vector (provenance, reputation, QoS, SLA_compliance, trust_val, timestamp)
16. construct trust_aware access model
17. generate access policy from access right generator

end

Online mode:

begin

1. construct workflow vector WF:= $\{t_1, t_2, t_3, \ldots, t_n\}$
2. S_i initiates service request SR // $S_i \rightarrow$ subject
3. start session
4. analyze SR
5. identify workflow instance (WFI) for subject S_i
6. set WFI:={subject, task set, session time}
7. collect context_info
8. identify trust_type
9. check trust_val
10. if trust_val =+1 then
11. S_i is honest
12. else if trust_val = 0 then
13. S_i is honest but curious
14. apply stronger Po // stronger policy
15. else if trust_val= -1 then
16. S_i is dishonest
17. block S_i
18. construct context_vector: = {context_info, trusted_relation}
19. update WFI based on context_vector
20. update access model
21. identify status (subject, designation, attributes) for WFI
22. identify expression and operation
23. set session_validity to duration
24. map WFI to access_model
25. bind (subject, designation, attributes, permission, object, operation, duration, context_vector)
26. dynamic_binding{(subject, designation), (designation, permission), (permission,object)}
27. if session_validity expires then
28. check WFI status
29. if WFI status= "finish" then
30. grant access
31. successful binding
32. WFI executes without interruption
33. else
34. deny access
35. binding not successful
36. block WFI

end

C. Mapping of Access Model to Workflow Instance

Workflow management module is mainly used to design workflow model to improve the efficiency of business process. Workflow instances are executed either in serial order or in parallel or in iterative way. It is based on a task set, where each task has a life cycle having states like ready, active, blocked, finished, and invalid. A mapping needs to be considered between workflow instance and access approach to determine the effect of workflow on access control. Figure 4 represents context and trust aware workflow oriented access approach.

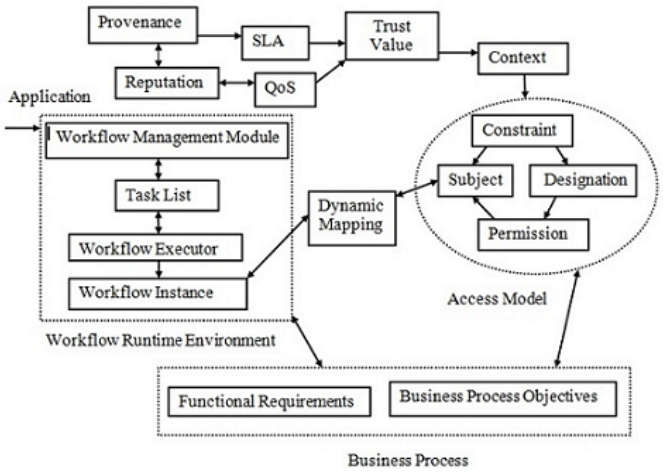

Figure 4. Context and Trust Aware Workflow Oriented Access Approach

Access dependency decides the control flow of workflow instance. As permissions are granted according to the current status of workflow instance, access control matrix (ACM) is extended by one dimension.

Therefore, $ACM_{dyn} : S \times O \times TWFI \rightarrow P(per)$

where $S \rightarrow$ subject, $O \rightarrow$ object, TWFI $\rightarrow$ task set for a workflow instance and P (per) $\rightarrow$ power set of all permissions.

It is assumed that subject S_i requests a specific permission Per_j for object O_k. ACM_{dyn} is used for this purpose to check current status of WFI for all active tasks.

- If Per_j entry is included within ACM_{dyn}, it is granted. If task t is not activated any more, then Per_j is revoked again.

- If Per_j entry is not included within ACM_{dyn}, then Perj is not granted. The extension from two to three dimensions increases the number of entries in the matrix.

The number of entries in $ACM_{dyn} := |S| \times |O| \times |TWFI|$

All permissions from ACM are also included in ACM_{dyn}. However, it can be said that ACM_{dyn} grants access more restrictively than ACM. The risk of data misuse in ACM_{dyn} is lower than in ACM. Therefore, it can be said that workflow oriented dynamic access control framework is more secure than others access control model.

V. ANALYSIS OF ACCESS FRAMEWORK

Workflow net is used here to represent access framework. The workflow specification represents a template, which is used to generate workflow instances by workflow executor. Order of execution of activities may be different at different instances. Workflow net model is defined and analyzed here by WF-net simulator WoPeD [11]. It has the following characteristics. Tasks in workflows correspond to transitions in workflow net. Executing a task corresponds to the firing of a transition. Marking of a workflow net represents the current state of a workflow. The control tokens represent the state of the workflow. The flow relation indicates how the tokens can move in the net. The control flow is determined by the flow relation and the start marking. The execution of workflows is supervised by the workflow executor and workflow management module to create and manage various instances of a workflow. Different sequences are created for every workflow instance. Access control is enforced upon the objects of the workflow instances.

In order to implement the access control mechanism, workflow management module generates and controls a workflow net for every instance. Every time a transition fires in an instance net, the marking has to be changed. Although we have examined proposed access framework at different scenarios to validate it at different workflow instances, we cannot present here the details of analysis due to limited space. It is sufficient to verify only a single instance to prove workflow net properties. We, therefore, present here only one instance. It needs to be determined whether a workflow net is completely executable or not, i.e., if the end marking is reachable from the start marking. This is analyzed by token game analysis and coverability graph. To check whether workflow specification is syntactically and semantically correct, concepts like liveness, deadlocks, and soundness are considered. Smooth execution of workflow instance is determined if source token reaches to terminal place successfully and all places are covered by the graph. This can also ensure consistency in proposed approach.

TABLE I. PLACES AND TRANSITIONS OF WORKFLOW NET MODEL

Place		Transition	
p1	initiate (with token)	t1	object_claimer
p2	service_request		
p5	workflow executor	t2	session
p7	context		
p8	trust	t4	WFI (workflow instance)
p11	constraint		
p12	granted	t14	access_decision
p13	denied		
p14	S-D binding (subject-designation)	t16	session_end
p15	D-P binding (designation-permission)		
p16	P-O binding (permission-designation)		
p19	terminate		

Table I represents places and transitions of workflow net model presented in figure 5. Qualitative analysis of this model by WoPeD tool shows it can satisfy workflow properties. It is well structured and can satisfy structural feasibility. This model is sound that can satisfy workflow net property, initial marking, boundedness and liveness. There are no wrongly marked places in the net models. As there are no unbounded place, boundedness properties are satisfied. As number of token in a place is limited to 1, it is bounded. It is also considered as a property of safeness. Conservation properties are also satisfied here as number of token remains 1 (constant) before and after execution. Liveness property checks the probability of deadlock. As there are no non-live or dead transitions, it can be said that the models satisfy liveness property too. Therefore the designed net model is considered as sound and safe. There are no probabilities of deadlock.

TABLE II. PHASE WISE FLOW-IN AND FLOW-OUT

Transition	Flow-In	Flow-Out	Phase
t1	p1	p2, p8	1
t2	p2, p8	p5, p11, p7	2, 3
t4	p5, p11, p7	p14, p15, p16	4, 5, 6
t14	p14, p15, p16	p12, p13	7
t16	p12, p13	p19	7

Table II represents flow-in and flow-out places associated with each transition of workflow instance presented in figure 5.

TABLE III. MARKING OF WORKFLOW NET COVERABILITY

From Marking	To Marking	Transition
1000000000000	0100100000000	t1
0100100000000	0011000100000	t2
0011000100000	0000000010110	t4
0000000010110	0000010000000	t14
0000010000000	0000000001000	t14
0000010000000	0000001000000	t14
0000000001000	0000000000001	t16
0000001000000	0000000000001	t16

Table III represents marking of workflow net coverability to show that all the places and transitions of the model can be covered and no two vertices have same marking. Therefore, it can be said that workflow net model is completely executable. The execution sequence of workflow instance in figure 5 is as follows.

sequence 1: {p1} → {t1}→ {p2, p8}→ {t2} → {p5, p7, p11}→{t4} →{p14, p15, p16} → {t14} → {p12} → {t16} → {p19}.

sequence 2: {p1}→ {t1}→ {p2, p8}→ {t2} → {p5, p7, p11}→{t4}→{p14, p15, p16} → {t14} → {p13} → {t16} → {p19}.

There are 5 transitions {t1, t2, t4, t14, t16} and 12 places {p1, p2, p5, p7, p8, p11, p12, p13, p14, p15, p16, p19}. In t14, AND-join-XOR split is used to combine (AND) the effect of the activities in S-D binding, D-P binding and P-O binding and then split (XOR) the flow either in p12 (granted) or in p13 (denied). In t16, XOR- join is used to either consider effect of p12 or p13 before termination.

Soundness property of workflow net model of proposed access control approach ensures that - there are no dead transitions, every transition state is reachable from source place to sink place triggered by a firing sequence, sink is the only place reachable from source place with at least one token in sink place.

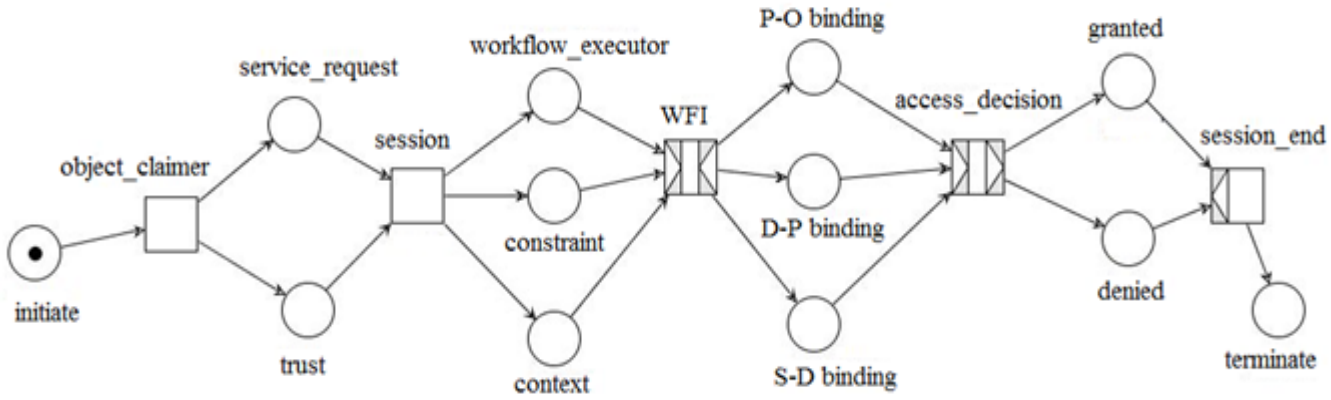

Figure 5. Validation of Context and Trust Aware Workflow Oriented Access Framework by WoPeD Simulator

VI. CONCLUSION

A major challenge in service-oriented computing environment is the design of effective access control approach. In this paper, a context and trust aware workflow oriented access framework is presented to solve challenges associated with business processes in Service Oriented Computing. General RBAC model cannot solve large number of temporary users. Therefore, RBAC model is not suitable for access control of SoC. Proposed approach is a flexible access control approach that binds subject to object in three phases- S-D binding, D-P binding and P-O binding based on inter-component relationship. Access model is mapped to workflow instance at run time. Validation of proposed framework is achieved by simulating through WoPeD tool. Soundness property of workflow net model guarantees the absence of livelocks, deadlocks, and other anomalies. After analyzing successful execution of various workflow instances as discussed in section V, it is established that there is no inconsistency and incorrectness in proposed dynamic access framework. It satisfies smooth execution of workflow instance without any interruption during analysis of model with token game approach.

Proposed approach is compared with workflow-oriented attributed based access control approach designed for SoC [12] to prove effectiveness of our approach. The work is going on to evaluate the performance of the proposed framework in healthcare domain, which cannot be presented here due to shortage of space. In the future, we will deploy this model and prove its validity.

VII. ACKNOWLEDGEMENT

This work is supported by Information Technology Research Academy (ITRA), Government of India under ITRA-Mobile grant (Ref. No. ITRA/ 15(59) / Mobile / RemoteHealth / 06).

VIII. REFERENCES

[1] Z. He, L. Wu, H. Li, H. Lai, Z. Hong, “Semantics-based access control approach”, Journal Of Computers, Vol. 6, No. 6, pp. 1152-1161, 2011.

[2] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, J. Reitsma, “Context sensitive access control”, In proceedings of SACMAT, pp. 111-119, 2005.

[3] J. Zheng, K. Q. Zhang, W. S. Zheng, and A. Y. Tan, “Dynamic role-based access control model”, JSW, 6(6):1096-1102, 2011.

[4] J. Lee, M. Winslett, J. Basney, and V. Welch., “The traust authorization service”, ACM Transactions on Information System and Security, VOL.11(1), 2008.

[5] H. Zhang, Y. He, Z. Shi, “Spatial context in role-based access control”, In proceedings of ICISC, pp. 166-178, 2006.

[6] B. Carminati, E. Ferrari, A. Perego, “Rule-based access control for social networks”, In proceedings of OTM Workshops, Springer, 2006.

[7] S. S. Yau, J. Liu, “A situation-aware access control based privacy-preserving service matchmaking approach for service-oriented architecture”, In proceedings of ICWS, pp.1056-1063, 2007.

[8] J. B. Bernabe, J. L.H. Ramos, A. F. S. Gomez, “TACIoT: multidimensional trust-aware access control system for the Internet of Things”, Soft Computing Journal, Springer, pp. 1-17, 2015.

[9] V. Atluri, W.K. Huang, “An Authorization Model for Workflows”, In Proceedings of European Symposium on Research in Computer Security, Springer, 1996.

[10] J. Zhu, Y. Zhou, W. Tong, “Access Control on the Composition of Web Services”, Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP), 2006.

[11] Workflow Petri net Designer, Available online at: http://woped.dhbw-karlsruhe.de/woped

[12] G. Zhanga, J. Liub, “ The Study of Access Control for Service-Oriented Computing in Internet of Things, International Journal of Wireless and Microwave Technologies, vol. 3, pp. 62-68, 2012.
Available online at http:// www.mecs-press.net/ ijwmt