IJCTA, 9(21), 2016, pp. 137-145

An Ontology Based Access Control Model for Social Networking Systems

Vipin Kumar* Parveen Kumar Amit Kumar Gupta*****

Abstract : In recent years, Online Social Networks (OSNs) have been achieved tremendous growth and become a de facto standard for social gathering for hundreds of millions of Internet users. Social gathering web sites or mobile apps such as Facebook and WhatsApp are naturally designed to enable people to share personal or public information and also make social relations with friends, colleagues, coworkers, family and even with strangers. These Social sites and Mobile Appsoffer attractive means for digital social interactions and information sharing, but also raise a big questionsof security and privacy. While its allow users to restrict access to shared data, but they currently do not provide any mechanism or trusted access control model to enforce privacy concerns over data associated with multiple users. Security and privacy are key concern for social networking systems. Traditional access control model were not so good, they were syntactic and error prone, lacking the necessary expressivity and efficiency of a solution where soundness and completeness of the underling logics is access control descriptions could be critical to harness their potential. In this paper, we have proposed an ontology based access control model for SNS. This model is for semantic web or Web 3.0. It can provide maximum security assurance with fast interaction among OSN users during online interaction. We also proposed ontologyfor this model to satisfy or validate the need of this framework using Protégé ontology development tools designed by University of Stanford and for result analysis we used Java based Jena Sematic Web Framework.

Keywords : SNS, Ontology, CWACM, OBACM, MABOGM, MABDBU

1. INTRODUCTION

Facebook, Twitter, &WhatsApp are essential parts of our life today. Role of these Online Social Networking sites are increasing very fast and involving as to participates or use these all social networking services, otherwise we will be far behind from modern era. These social networks are just simple graphs with nodes for the people and groups. These groups of people links for the relationships. In practice, the links can encode all kinds of relationships – familial and friendship. Online communities like Facebook, Twitter &WhatsApp are groups of people connected. These communities have become an important part of modern society and contribute to life in many contexts – social, educational, political and business. The infrastructures & communication technologies used to support online communities like Facebook, Twitter &WhatsApp have evolved with the Internet and include electronic mailing lists, bulletin boards, UseNet, IRC, Wikis, and blogs. These online communities built on social network structures began appearing from 2002 and have become most popular web-based applications now a day's. Such social networking sites allow individuals to publish personal information in a semi-structured format and to define links to other members with whom they have relationships of various kinds. In recent years, there have been two parallel areas in access control research. On one hand, many researchers are efforts to develop access control models to fulfill the authorization requirements from real-world application domains. These have turned out several successful and well-established access control models, such as NIST/ANSI standard RBAC model [1, 2], the RT

* Ph. D Scholar of NIMS University, Jaipur, Rajasthan, Department of Computer Application, *KIET, Ghaziabad*

** Department of Computer Application KIET, Ghaziabad, Computer Science & Engineering, NIMS, University, Jaipur, Rajasthan

*** Department of Compute Application, KIET, Ghaziabad.

model [3], and the Usage Control model [4]. On other hand, in parallel, and almost separately, many researchers have devoted to develop policy languages for access control to support policy-based computing, including application-level policies (*e.g*.SAML [5], Ponder [6], XACML [7], and EPAL [8]), network-level policies (*e.g*. IPSec policy[9] &firewall policy [10]), and system level policies (*e.g*. SELinux policy [11]).

But, there is big gap between access control developing and implementing it on real life application like social networking system or cloud computing, because there is not any standard framework to implement these access control model in these application. In this paper, we have proposed an ontology based access control model for Social Networking Systems (SNS). This access control model is based on third generation web 3.0 or semantic web.

This paper is divide into several sections, section II for existing access control model that is using currently for social networking systems, section IV for propose access control model representation, section V for implementing ontology for proposed access control model, section VI for result analysis and finally section VII for conclusion and future scope.

2. CURRENTLY WORKING ACCESS CONTROL MODEL(CWACM)

Social networking models can be represents differently by different point of view, but in this paper we are representing most common currently working access control model in figure 1.In this model user is the actor that make request usingweb browser or by using any user interface on Internet. In this model web browser is the user interface that used by user or actor to interact with SNS. By this way, we usually interact with social networking systems.

Here, by SNS we mean social networking websites like Facebook and Twitter. These SNS web sites directly make connection with database managed by DBMS or RDBMS using 2-tier or 3-tier architecture of networking. This interaction between database layer and SNS may be implemented using Client/Server architecture or distributive DBMS architecture

Last layer of this model represents the database management system for storing data in SNS. This data may be store in wide column stores/column family databases like Hadoop/Hbase, or may be store in document store database like MongoDB or may be store in key value/tuple store database like Amazon DynamoDB or may be store in SQL based database management software like MSSQL, Oracle, and MySQL etc.

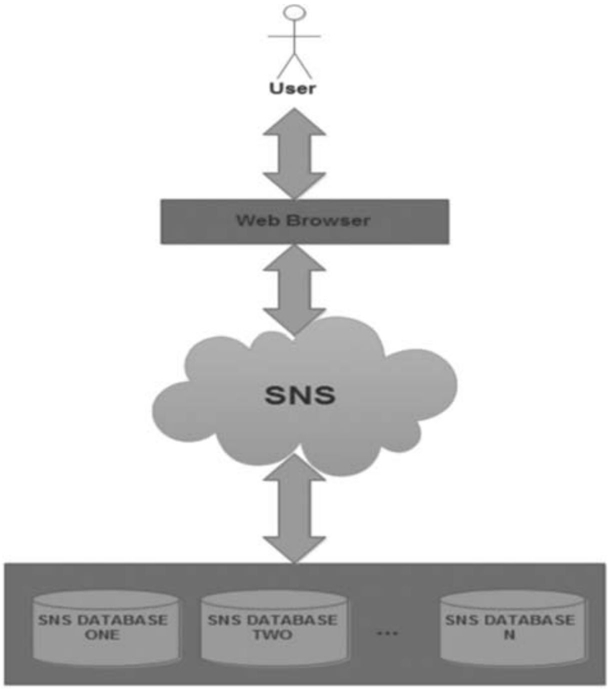

Fig. 1. Currently Working Access Control Model

This currently working access control model approach is very simple or straight forward that have less flexibility or security. This model is more suitable for second generation Web 2.0, but may not be suitable for third generation Web 3.0 or Semantic Web. In this type of access control model database access type is more or constant for each and every transaction with the database. In any case, if load on server is very high then it takes more time for transaction as compare to normal time taken for each transaction.

In this paper, we purposed a new innovative an ontology based access control model for SNS, which is more suitable for third generation web 3.0 or semantic web environments. If this access control model implemented carefully then, it will take less time for transaction as compare to currently running ACM.

3. PROPOSED ONTOLOGY BASED ACCESS CONTROL MODEL (OBACM)

This OBACM model is completely based on innovative idea that came in mind after studying related work in OSN and understanding current working environment of OSN. This model is distributed in two sections, first section control or interacted by users or actors and on other hand second section is control or interacted by administration of this model.

In the first section, the first layer is User. User is also actor in this modelas it was in CWACM model that interacts with the other OSN users via web browser that may be friends, relatives or unknown users of same SNS on Internet.

In first section second layer is Query Analyzer. Query Analyze is the interface that gets requests from users interface via Internet and sends request to Multi-Agent Based Ontology Manager in sentence skeleton form. On the base of knowledge based available for understanding sentence skeleton send by query analyzer, MABOM generate SPARQL query that can be understand by SPARQL engine using query generator and pattern heuristic and then send that query to SPARQL engine.

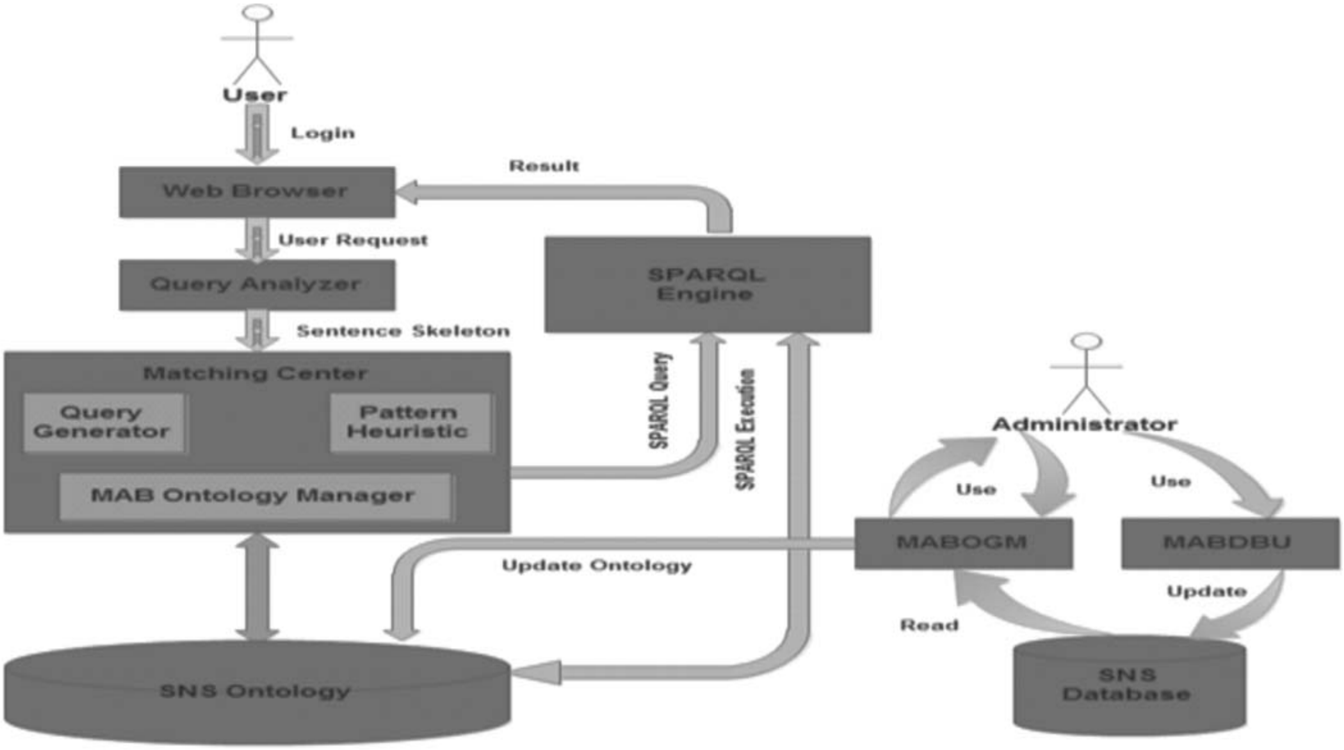

Fig. 2. Proposed OBACM Model or Framework

In same first section, SPARQL engine play big role to produce result on the basis on knowledge available in SNS Ontology. SPARQL is a recursive acronym for the SPARQL Protocol and RDF Query Language. SPARQL is a structure query language for RDF as SQL is a structure query language for DBMS or RDBMS. This SPARQL engine execute SPARQL query send by MABOM using knowledge based available in SNS Ontology and send result of that SPARQL query to the user or actor that made request earlier for the specific query.

In first section last layer is SNS Ontology. This SNS Ontologyuse Ontology. Ontology is a data model for semantic web that represents knowledge as a set of concepts within a domain and the relationships between these concepts. In short Ontology is a data management for semantic web as Oracle or MySQL are that why, here SNS Ontology represent semantic web ontology that have assertions, relationship, rules and regulation among classes, object properties, data properties and individual.

In second section, Administrator is the actor in that interacts with the SNS Database in this model using MABOGM and MABDBU. MABOGM(Multi-Agent Based Ontology Generator and Modifier) is AI application that managed by administrator to create or update SNS ontology on time to time bases as per new rules inserted in SNS Database. MABDBU(Multi-Agent Based Database Updater) is AI application that managed by administrator to update or modified SNS Database on time to time bases as per requirement to make database up to date.

In second section, SNS Database may be SQL or NOSQL based database management system that store data related to elements and relations, such as a set of users, a set of roles, a set of permissions, and relationships between users, roles, permissions and etc.

4. MAKING OF SNS ONTOLOGY

4.1. SNS ontology implementation

In this paper, we use Protégé [12] for creating ontology.This proposed SNS Ontology has key entities and their relationships that typically found in SNS. This SNS Ontology is the version 3 that comprises of 22 classes and 52 object properties that is upgraded from version of SNS ontology version 2[13,14]. Figure 3 shows graphical representation of relationship among classes in SNS Ontology.

In this SNS Ontology, Thing is the root class, with eight immediate descendants classes are:DigitalObjects, Persons, Relations, Events, Resources, Owners, Roles, andPermission

Use of these classes as follows :

1. The DigitalObjects class stores object with digital properties. The Digital Objects class is specialized by sub-classes such as Photos, Wall, Notes, VideoClips and Annotation.
2. The Persons class stores human users in the context of Social Networking and it is specialized by Woman and Man class. Man class represents male and Woman class represent female.
3. The Notes class use to store textcontent or data.
4. The Wall class models the posting board on the homepage of a currently login person, such as the one Facebook or Orkut provides.
5. The Annotation class represents special digital objects that instead of directly representing a content, annotate one object (*e.g*., a wall, a photo, etc.) using another object (*e.g*., a textual comment, a person, etc.). The two objects are related to an annotation object, using properties Annotates and AnnotatesWith, respectively.
6. Annotation class specialized by Comment, Tag, and WallPost.
7. Comment class used to annotates an object with a note.
8. Tag class specialized by PersonTag.
9. PersonTag class specialized by two subclasses PhotoPersonTag and VideoPersonTag.
10. PhotoPerson Tag class is a specialized tag that annotates a photo with a person
11. VideoPerson Tag class is specialized tag that annotates a video with a person.
12. WallPost class annotates a wall with an object, *e.g*., a photo or video.
13. Relation class represent the relationships among users.
14. Events show individual or group events details
15. Resources class used to store recourse owner information
16. Owners class is used to classified the owners information on individual objects or digital objects.

17. Roles class is used to represent the roles or rights of individual users on SNS.
18. Permission class used to store read, write, execution, postby or shareby information of objects.

This figure 3 show overall is-a relationships of all classes implemented in this SNS Ontology using Protégé Ontology Manager application.

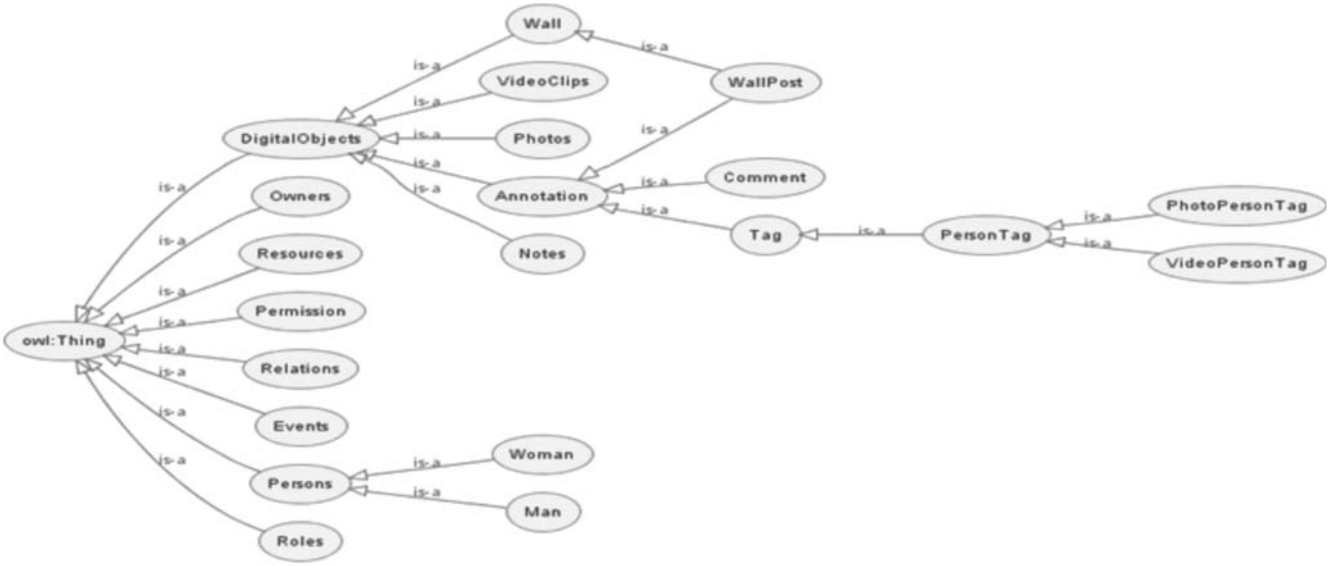

Fig. 3. SNS Ontology Classes.

4.2. SNS ontology rules and use case

Semantic Web Ontology use SWRLfor accomplishedrules for Access Control, in order to infer new knowledge which does not exist in the Ontology knowledge base. The first issue is the translation and use of the knowledge base developed in OWL in SNS Ontology that is generated using Protégé. The OWL triples are easily transformed into SWRL facts. These rules can be used in the inference process as the fact base.

In this section, we have carefully worked on generating rules and knowledge base for SNS Ontology. We have also described the overall logical access control model for intelligent access control decisions.

For generating or building up new knowledge base resources are needed to be captured or store using ontology. In SNS, access control policies need to be semantically expressingthem using ontology concepts. Our approach is to capture the other components of policies for getting right decision using ontology. By this way, access control policies or protected knowledge resources can be naturally integrated to facilitate an efficient and semantic rich access control decision in access control framework.

For this purpose, we propose SNS ontology for access control model for secure, optimized and fastest social networking systems. In our Ontology, We consider relations main protection objects in an ontology-based knowledge base. However, current semantic web based languages like OWL do not support such kind of statement about the relation instances, which is necessary for specifying authorizations on them. In order to support this, we have modified some rules using Java language and tried to implement them on Jena Semantic Web Framework as much as possible to implement. There are so many approaches available for it like reified [15], personal authorization, permission authorizations,basic authority specification, direct authorization, dependent authorization, multi-authority specification and dependent authorization. But we do not consider all of them in our ontology policies; we have categories all of them in two types direct or indirect inference rules.

Direct Rules: In this SNS Ontology, we have asserted more than 100 direct rules as prototyped for running or testing this access control model. These direct authorization rules for ontology allow the access control system to grant permissions to users or actor without involvement of user authorities. Similar to a personal authorization rule, the antecedent specifies the protection resource. The prototyped formats of direct authorization rules used in this SNS Ontology are shown in table 1 as follows:

Table 1. Direct Rules in SNS Ontology

S. No	***Prototyped Direct Rules in SNS Ontology***
1.	sns:property(comment 4, photo1) ⇒ sns:postedOn(comment1, photo1)
2.	sns:property(alice, marry) ⇒ sns:is Friend Of (alice, marry)
3.	sns:property(shashank, amit) ⇒ sns:has Friend (shashank, amit)
4.	sns:property(arti, preveen) ⇒ sns:is Sister Of (arti, preveen)
5.	sns:property(ankit, vipin) ⇒ sns:is Friend Of (ankit, vipin)
6.	sns:property(note1, photo1) ⇒ sns:annotates With (note1, photo1)
7.	sns:property(alice, robert) ⇒ sns:has Brother(alice, robert)
8.	sns:property(robert, video1) ⇒ sns:can View(robert, video1)
9.	sns:property(gita, sita) ⇒ sns:is SiblingOf(gita, sita)
10.	sns:property(sachin, vipin) ⇒ sns:is BrotherOf(sachin, vipin)
11.	sns:property(robert, tom) ⇒ sns:is FatherOf(robert, tom)
12.	sns:property(vipin, comment4) ⇒ sns: posted(vipin, comment4)
13.	sns:property(vipin, admin) ⇒ sns: roleOf(vipin,admin)
14.	sns:property(shani, photo1) ⇒ sns: canRead(shani,photo1)

These above ontology rules state that a subject can read a property instance of which it is either the object or subject respectively.

Indirect Rules : In SNS Ontology, we have asserted more than 100 indirect rules as prototyped for running this access control model. These indirect rules for ontology are extracting from combination of one or more direct rules and extract with the help of semantic web languages like SWRL etc. SWRL rule contains antecedent part and a consequent part. If all the atoms in the antecedent are true, then the consequent must be true. Some of the main indirect rules enforced in this ontology are follows in table 2.

Table 2. Indirect Rules in SNS Ontology

S. No	***Indirect Rules in SNS Ontology***
1.	hasParent(?a, ?b) ^ man(?b) – > hasFather(?a, ?b)
2.	hasSibling(?a, ?b) ^ woman(?a) – > hasSister(?a, ?b)
3.	hasChild(?a, ?b) ^ woman(?a) – > hasDaughter(?a, ?b)
4.	hasParent(?a, ?b) ^ hasSister(?a, ?c) – >hasAunt(?a, ?c)
5.	Person(?a) ^ has Age(?a, ?age) ^ swrlb:greaterthan(?age, 17)– > adult(?a)
6.	hasChild(?a, ?b) ^ man(?a) – >hasSon(?a, ?b)
7.	hasParent(?a, ?b) ^ woman(?a) – >hasMother(?a, ?b)
8.	Person(?a) ^ hasSibling(?a, ?b) ^ man(?b) – >hasBrother(?a, ?b)
9.	hasSibling(?a, ?b) ^ man(?a) – >hasBrother(?a, ?b)
10.	hasParent(?a, ?b) ^ hasBrother(?b, ?c) – >hasUncle(?a, ?c)
11.	isFriendOf(?a,?b)^ uploadedBy(?a,?c) – >canRead(?a,?c)
12.	ownerOf(?a,?b)^ postedBy(?c,?b) ^man(?c)^man(?a) – >hasFriend(?a,?c)

These ?*a* and ?*b* variables represent the object and subject represented in this SNS Ontology. These indirect rules are derived from one or more direct rules to make ontology more dynamic or knowledge oriented.

4.3. Query Analysis

In order to infer new knowledge which does not exist in the knowledge base, SWRL language is used to accomplished access control. The first issue is the translation and use of the knowledge base developed in OWL. The OWL triples are easily transformed into SWRL facts. These rules can be used in the inference process as the fact base. In our implementation of SNS Ontology, rules are written and executed as Jena Rules.

Let take an example to understand its rules, a user name Robert wants to open Video1. We hereby need to describe formally the following premises:

1. User “Robert” is person, and, “Robert” is the Owner of “Video1”
2. The Role of user Robert is “Administrator”.
3. As an “Administrator” Robert has a Read, Write and Watch permission on Video1.

The logical characterization would be as follows:

(“Robert” is Member Of “Person) &&(“Robert” **isOwner Of**” Video1") && (“Robert” **has Role Of**”Admin”) && (“Robert” **has Permission Of**”Read”) → (“Robert” **canWatch** “Video1”)

Same rule can be express in Jena Rule Format as follows:

@prefix ont: <URI_ONTOLOGY#>.

@include <RDFS>.

[rule_Video1_NOT_REST_RESOURCES: (?*i* ont:isOwnerOf ?*x*) notEqual(?*x*, ont:Video1) – > (?i ont:hasNegResources ont:Video1_NOT_RESOURCES)]

[rule_Video1_RES: (?iont:results ont:Video1) < – (?*i* ont:hasResource ont:Video1) noValue (?*i*, ont: has Neg Resourcesont: Video1_NOT_RESOURCE)]

[rule_Video1_NOT_REST_USERS: (?*i* ont: **itsOwnerIs**? *x*) notEqual(?*x*, ont:Robert) – > (?*i* ont:has NegUsers ont:Video1_NOT_USERS)]

[rule_Video1_USERS: (?iont:results ont:Video1) < – (?*i* ont: **its OwnerIs**ont:Robert) noValue(?*i*, ont: has Neg Usersont: Video1_NOT_USERS)]

[rule_Video1_NOT_REST_ROLES: (?iont: **hasRoleOf**?*x*) notEqual(?*x*, ont:ADMIN) – > (?*i* ont: has Neg Roles ont:Video1_NOT_ ROLES)]

[rule_Video1_USERS: (?iont:results ont:Video1) < – (?*i* ont: **has RoleOf**ont:ADMIN) noValue(?*i*, ont: has Neg Rolesont: Video1_NOT_ ROLES)]

[rule_Video1_NOT_REST_PERMISSIONS: (?*i* ont: **permission** ?x) **notEqual(?*x*, ont:READ) notEqual(?*x*, ont:WRITE)** – > (?*i* ont: hasNegPermissions ont:Video1_NOT_ PERMISSIONS)]

[rule_Video1_PERMISSIONS: (?*i* ont:results ont:Video1) < – **(?*i* ont: permission ont:READ) (?*i* ont: permission ont:WRITE)** noValue(?*i*, ont: has Neg Permissionsont: Video1_NOT_ PERMISSIONS)]

[rule_Video1_NOT_REST_RESBIS: (?*i* ont: **resource** ?*x*) **notEqual(?*x*, ont:Video1)** – > (?*i* ont: hasNegResBis ont:Video1_NOT_ RESBIS)]

[rule_Video1_RESBIS: (?iont:results ont:Video1) < – **(?*i* ont: resource ont:Video1)** noValue(?*i*, ont: hasNegResBisont: Video1_NOT_ RESBIS)]

Similarly, we can convert more SWRL rules into Jena format rules to test this Ontology on Java Runtime Environment.

5. RESULT ANALYSIS

This test has been conducted for access control engine by submitting SPARQL queries on Jena Semantic Web Framework. The engine successfully returns only the authorized information that is expected according to the sample access control policy rules. We also developed a data generator that randomly populates SNS ontology rules on Jena Framework and generates results. The performance results of the prototype access engine based on the following input parameters are shown in Table1: the number of users, number of friend's links, numbers of photos, and maximum number of people had been tagged in a photo. Since the inference engine that Jena provides only works in memory, we were not able to run the experiment for very large ontologies.

In our experiments, it is also clearly shown that the first access control inference is relatively expensive. However, subsequent access checks are performed almost instantaneously. This is because in the first round the inference model caches some of the inferred axioms, which enhances performance for subsequent inference. In fact, the first access check can be considered as part of the initialization phase, which can be triggered with a dummy access request.

Table 3. Prototype of Performance Results

Data Generation Parameters				*Access Times (in second)*		
User	*Photo*	*Tag/Photo*	*isFriendOf*	*Initial*	*First*	*Subsequent*
10	5	5	50	2.0	0.4	0.035
50	25	20	125	4.2	32.0	0.042
80	60	20	180	6.23	176.25	0.066
115	70	22	213	14.4	2126.2	0.009
143	90	34	503	21.3	4344.8	0.007
205	110	56	602	22.5	5421.6	0.008

This result may be change on different environment or condition.

6. CONCLUSION AND FUTURE SCOPE

In this paper, we have proposed an innovative access control model base on ontology for SNS. That model is capable to run faster and still is secure as compare to other existing access control model. The key idea behind in this model is to express the policies on the relations among concepts in the social network ontology. We have also implemented this model prototype of the proposed model in order to show the applicability of our approach. Although this model provides powerful access control features to the users of the SNSs, even savvy users of such systems should not have to be able to compose access control policy rules manually. An SNS employing this framework may simply provide a user interface similar to the current practices, but with more flexible options to its user; then, provide the access control engine with policy rules corresponding to the user choices.

In future research, this model is tested in virtual environment, which is not accurate after implementation on real environment. But so far, fewer resources are available for semantic web to implement this model on real environment. We have tested this model on Jena Semantic Web Framework, but it is not 100% compatible for purposed access control policies, in future one may try to test it on real time environment or more semantic web compatible framework that may be developed in the future.

7. REFERENCES

1. D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Trans. Inf. Syst. Secur. (TISSEC)*, vol. 4, no. 3, pp. 224–274, 2001.

2. ANSI, American National Standards Institute Inc., Role Based Access Control, ANSI-INCITS 359–2004, 2004.

3. N. Li, J.C. Mitchell, and W.H. Winsborough, “Design of a role-based trust management framework,” in Security and Privacy, 2002. Proceedings.2002 IEEE Symposium on. IEEE, 2005, pp. 114–130.

4. J. Park and R. Sandhu, “The UCON ABC usage control model,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, pp. 128–174, 2004.

5. OASIS, “Security Assertion Markup Language,” http://www.oasis-open. org/committees/security/.

6. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “The ponder policy specification language,” *Policies for Distributed Systems and Networks*, pp. 18–38, 2001.

7. XACML, “OASIS eXtensible Access Control Markup Language (XACML) V2.0 Specification Set,” http://www.oasis-open.org/committees/xacml/, 2007.

8. P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, “Enterprise privacy authorization language (EPAL),” http://www.w3.org/Submission/ 2003/SUBM-EPAL-20031110/.

9. M. Condell, C. Lynn, and J. Zao, “Security policy specification language,” *Internet Engineering Task Force (IETF) Internet Draft*, 2000.

10. D.B. Chapman, E.D. Zwicky, and D. Russell, *Building internet firewalls*, O’Reilly & Associates, Inc. Sebastopol, CA, USA, 1995.

11. P. Loscocco and S. Smalley, “Integrating flexible support for security policies into the Linux operating system,” in *Proc. 2001 USENIX Annual Technical Conference REENIX Track*, 2001, pp. 29–40.145

12. http://protégé.standford.edu

13. Vipin Kumar, and DrSachin Kumar, “Access Control Framework for Social Network System using Ontology”, *International Journal of Computer Applications (0975 – 8887) Volume 79 – No4, October 2013*

14. Vipin Kumar, Dr. Sachin Kumar, “Multi-Agent Based Access Control Framework for Social Networking Systems Using Ontology”, *International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 2, Issue 1, January 2014*

15. AmirrezaMasoumzadeh, and James Joshi, “Ontology-based access control for social network systems” in *Int. J. Information Privacy, Security and Integrity, Vol. 1, No. 1, 2011*