Policy Specification and Enforcement in Online Social Networks using MKNF$^+$

Mahdi Alizadeh, Seyyed Ahmad Javadi, Morteza Amini, Rasool Jalili
Data and Network Security Lab (DNSL)
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
{malizadeh@ce., ajavadi@ce., amini@, jalili@} sharif.edu

Abstract—**Emerging tools that ease sharing information in online social networks (OSNs) can cause various privacy issues for users. Access control is the main security mechanism in OSNs which is used to tackle such issues. In this paper, a prioritized ontology based access control model for protecting users' information in OSNs is proposed. In the proposed model, description logic (DL) is used for modeling social networks and MKNF$^+$ rules are used for specification of users' access control policies. Using MKNF$^+$, we can have nonmonotonic inference (i.e., closed-world reasoning) in the access control procedure. Conflict among access rules defined by a user in an OSN, is another problem, which is resolved in the proposed model by defining priority levels for the rules in a logical manner. Logical foundation of the model gives accuracy, expressiveness, and inference (of implicit access rules from the explicit ones) to the model, and thus decreases the risk of sharing information in OSNs.**

***Keywords-** Access Control, Logic, Online Social Network, Semantic Web*

I. Introduction

Users in Online Social Networks make profiles and share different types of resources such as their personal information, photos, notes, and videos with others. In addition, users can establish relationships with other users and communicate with them. In fact, OSNs are built for different purposes. Some OSNs are general purpose social networks and they help users to communicate with their friends better. In contrast, some OSNs are built for a specific purpose such as business or art. On the one hand, users prefer to expand their social activities and share more content with others, however, making these contents available to others raises some privacy issues for users.

Traditional access control models such as discretionary, mandatory, or role-based access control cannot cover whole the social networks' requirements. In OSNs, there is no central authority in the system and users themselves determine which groups of users are authorized to access their resources. Defining a list of authorized users for each resource is a cumbersome job for users. Furthermore, OSNs are dynamic environments. For instance, users change their relationships with others dynamically. Thus, if users would like to make a list of authorized users accessing to their resources, they should update these lists continually. These features distinguish access control in OSNs from other environments. In OSNs, instead of making a list of authorized users for each resource, properties of users and resources are used for deciding whether a request should be permitted or not. To illustrate, users can use their relationships to define which users can access their recourses.

Current OSNs generally use simple parameters to protect users' resources. Relationships that are established between users usually are considered as the most important parameter for protecting resources. For instance, in Facebook [1] "everyone", "friends only", and "friends of friends" can be used as parameters for determining which users are authorized to access specific types of resources.

In this paper, a prioritized ontology based access control model for OSNs is proposed. In the proposed model, Description Logic (DL) is used to describe concepts, roles, individuals, and relationships among them. Since description logic does not support rules, in our proposed model, MKNF$^+$ [1] is used as a combination of description logic and Answer Set Programming (ASP). Using MKNF$^+$ in the proposed model has some advantages. First of all, it increases the expressive power of description logic by incorporating rules. Secondly, arity restriction does not exist in ASP predicates and this makes access rule definitions easier in logic. Third, using MKNF$^+$ makes it possible to have some nonmonotonic features such as specification of default rules and closed-world reasoning.

The rest of the paper is organized as follows. Section II reviews the proposed models for access control in OSNs. In Section III, an overview of MKNF$^+$ is described briefly. In Section IV, a simple OSN is modeled. In Section V, our proposed access control model for OSNs plus approaches for conflict resolution are introduced. In addition, a case study for clarifying applicability of the model is mentioned. Finally, Section VI concludes the paper and draws some future directions.

II. Related Work

Preserving users' privacy and enforcing access control in OSNs are interest of many researchers. Boyd and Ellison [2] mention some privacy and security issues of OSNs. Carminati *et al.* [3] proposed a semi-decentralized access control model for these environments. To be assured that enforcement of

[1] http://www.facebook.com

978-1-4673-2386-4/12/$31.00 ©2012 IEEE

access control is carried out accurately, they proposed to use client-side access control. In the proposed model, the relation type, distance, and trust among users are considered as parameters that can be used for defining access control policies. Fong *et al.* [4] formalized the access control mechanisms behind Facebook. In addition, they mentioned how these policies can be extended. Carminati *et al.* [5] proposed to use semantic web tools for enforcing access control policies. Authorization, admin, and filtering policies are mentioned as different types of policies which can be defined by users in OSNs. These policies are modeled by OWL and Semantic Web Rule Language (SWRL). Masoumzadeh and Joshi [6] considered protecting relationships among concepts in their proposed model. They mentioned that users should have ability to control visibility of their relationships; furthermore, the proposed model supports defining multi-authority access policies. In comparison to these models [5] [6], we propose to use MKNF$^+$ instead of SWRL for defining access policy rules. SWRL is a kind of Horn-like rules. It is a combination of OWL and the unary/binary Datalog Rule Markup Language. In MKNF$^+$, we can define predicates with arbitrary arities. Furthermore, MKNF$^+$ supports nonmonotonic inference rule such as negation-as-failure. SWRL is a monotonic logic because its semantics is defined in the first-order language.

Additionally, various models have been proposed for conflicts detection and resolution among access control rules. Bertino *et al.* [7] proposed to define weak and strong authorizations. In this model, strong authorizations have more priority than weak authorizations. Cuppens *et al.* [8] proposed an Organization Based Access Control (OR-BAC) which supports conflict detection and resolution. They restricted structure of rules which users can define for preserving decidability. To the best of our knowledge, no prioritized ontology based access control model has been proposed for OSNs yet.

Several frameworks have been proposed for combining DLs and rules. AL-log [9] combines ALC with positive Datalog programs. DL-log [10] is proposed to extend AL-log. DL-log supports disjunctive Datalog with negation. In addition to unary predicates, in this framework, binary predicates are supported too. Such frameworks use DL-safety condition as a restriction for integration of ontology and rules. According to the DL-safeness condition, each variable in a rule should occur in a non-DL atom in the rule's body. This restriction affects the expressive power of the framework. CARIN [11] is a framework which does not follow safe interaction between DL and rules. In fact, unrestricted interaction among DL and rules would limit expressiveness of at least one of DL or rules. Donini *et al.* [12] used autoepistemic operators and added negation-as-failure to DL ALC. In this paper, we use MKNF$^+$ as a formalism which combines DL and rules. In comparison to proposed formalisms, to the best of our knowledge, MKNF$^+$ is the most powerful decidable formalism proposed for combination of description logics with rules.

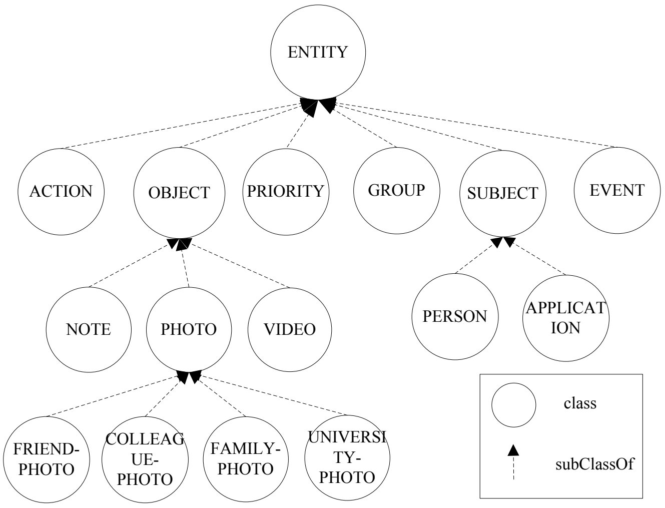


Figure 1. The Proposed Ontology for Online Social Networks

III. Introduction to MKNF$^+$

Description logic is a strong language for knowledge representation. However, some syntactic restrictions are considered for preserving decidability of DL. For example, specific rules such as "an uncle is the brother of one's father" cannot be expressed in DL [1]. Moreover, nonmonotonic reasoning is not supported by DL.

MKNF$^+$ is a formalism proposed for combination of DL and ASP. In this formalism, DL-predicates are defined in the DL part of the language and other predicates are called non-DL-predicates. In contrast to non-DL-predicates, the arity of DL-predicates is bounded. In fact, these predicates should be unary or binary predicates. Moreover, two types of modal-atom namely K-atom and not-atom are defined in this formalism. The structure of an MKNF$^+$ rule is as follow:

$$B_1, ..., B_n \rightarrow H_1 \vee ... \vee H_m \quad (1)$$

In Rule (1), B_i can be a non-modal, a K-atom, or a not-atom, whereas, H_i is a nonmodal or a K-atom. For preserving decidability of MKNF$^+$, DL-safety restriction is defined. According to this restriction, each variable in a rule should appear in the body of the rule in some non-DL-atom which is restricted by the K operator.

IV. Online Social Network Model

Nowadays, the emergence of OSNs with diverse purposes, urges users to join and share their information in these sites. In this paper, we do not intend to propose an ontology which models all the concepts and their relationships in various OSNs. In fact, a simple ontology for modeling general OSNs is proposed for proving applicability of the proposed access control model for OSNs. In addition, this ontology can be easily extended to support other OSNs' requirements. The proposed ontology is represented in Fig. 1. Users usually prefer to define different access control policies for various types of resources which they share in OSNs. To illustrate, a user may tend to share his/her notes with his/her friends, but

TABLE I. Online Social Network Relationships and Non-DL-Concepts

Type	Predicates

DL-Relationships	IS-FRIEND-OF(*Person*, *Person*)
	IS-CLASSMATE-OF(*Person*, *Person*)
	IS-FAMILY-OF(*Person*, *Person*)
	IS-CHILD-OF(*Person*, *Person*)
	IS-PARENT-OF(*Person*, *Person*)
	IS-MEMBER-OF(*Person*, *Group*)
	IS-COLLEAGUE-OF(*Person*, *Person*)
	IS-MANAGER-OF(*Person*, *Person*)
	IS-EMPLOYEE-OF(*Person*, *Person*)
	ATTEND-IN(*Person*, *Event*)
	USE-APPLICATION-OF (*Person*, *Application*)
	OWNS(*Person*, *Object*)

Non-DL-Concepts	o(*Object*)
	s(*Person*)

TABLE II. Access Control Predicates

Type	Predicates

Basic Authorization	permit(*Authority*, *Subject*, *Action*, *Resource*, *Priority*)
	prohibit(*Authority*, *Subject*, *Action*, *Resource*, *Priority*)

Priority Enforcement	h-permit(*Authority*, *Subject*, *Action*, *Resource*, *Priority*)
	h-prohibit(*Authority*, *Subject*, *Action*, *Resource*, *Priority*)
	hasMorePriority(*Authority*, *Priority* , *Priority*)

Final Decision Authorization	fd-permit(*Authority*, *Subject*, *Action*, *Resource*)
	fd-Prohibit(*Authority, Subject, Action, Resource*)

he/she is not inclined that his/her photos are accessed by them. In addition to the type of a resource, the topic or the context of the resource is an important parameter that can be used for determining which users are authorized to access a resource. For example, suppose a user would like to share his/her university notes with his/her classmates, but he/she is not inclined to share them with his/her colleagues. In the proposed ontology for OSNs, various sub concepts are defined for each object. Due to the lack of space, just photo sub concepts are represented in Fig. 1. Subjects are categorized to users and applications. Nowadays, developers can create new applications by using API's provided by OSNs. These applications can access users' profile and provide special services for them. Users should able to control witch part of their information can be accessed by these applications. Various types of groups and events are defined by users in OSNs. Concepts "GROUP" and "EVENT" are considered in the ontology for representing these groups and events. Furthermore, users can define various priority levels for their access rules. These priority levels are defined as individuals of concept "PRIORITY". Finally, by using concept "ACTION", various actions which users can do in OSNs are modeled.

Provided that an OSN supports various types of relationships, users can define their relationships more accurately. Consequently, they will be able to define better access control rules for protecting their resources. Moreover, in this paper, these relationships are modeled as directed labeled edges in the social network graph. In the proposed model, description logic is used for describing the subsumption hierarchy of the concepts. Relationships considered among concepts are listed in TABLE I. In addition, two types of predicates are defined, namely, description logic predicates (which are represented by *uppercase* names in the paper) and non-DL-predicates (which are represented by *lowercase* names in the paper). All the users and resources shared in the OSN are defined as members of non-DL-concepts *"s"* and *"o"* respectively. For example, if Alice shares a new note called NOTE$_1$ with her friends, NOTE(NOTE$_1$) and o(NOTE$_1$) are added to the knowledge base. By doing so, the DL-safety restriction is satisfied in access control rules.

V. ACCESS CONTROL MODEL

Designing an appropriate access control model for protecting resources against unauthorized access can be regarded as the first step for protecting users' privacy in OSNs. In fact, in this section, various access control rules will be defined using MKNF$^+$. Since positive and negative authorizations are supported in defining access control rules, conflict among rules is possible. Our proposed approaches to resolve these conflicts are mentioned in this section.

A. Policy Specification

An access policy rule can be divided into an antecedent and a consequent. If conditions in the antecedent of a rule are satisfied by the knowledge base, predicates in the consequent of the rule will be added to the knowledge base. Unlike DL predicates, non-DL-predicates do not have any arity restriction. Hence, we use non-DL-predicates for defining predicates with arbitrary arity in our access control rules. In the proposed model, users can define positive and negative authorizations for protecting their resources. These predicates are called basic authorization and are represented in the first row of TABLE II. Users can permit or prohibit groups of users to access a resource. A user who grants permission, a user who takes the permission, an action requested on the resource, requested resource, and the priority of this predicate are parameters of *permit* and *prohibit* predicates. If priorities of these two predicates are not assigned by a user in an access control rule, default priority is considered for these predicates. Various actions namely *create*, *delete*, *read*, and *share* are considered as the actions that users can do when they are in OSNs. Such actions are added to the knowledge base as individuals of "ACTION". For instance, if Alice tends to let third party applications to access her photos, she can define Rule (6). Moreover, she assigns high priority to this rule. This rule satisfies DL-safety restriction because "rsc" and "sbj", which are our variables, occur in "o" and "s", which are non-DL-predicates in the antecedent of the rule, respectively.

$$\begin{aligned}&\mathbf{K}\,\mathrm{o}(?\,\mathrm{rsc}), \mathbf{K}\,\mathrm{s}(?\,\mathrm{sbj}), \mathbf{K}\;\mathrm{PHOTO}(?\,\mathrm{rsc}),\\ &\mathbf{K}\,\text{USE-APPLICATION-OF}(\mathrm{Alice}, ?\,\mathrm{sbj})\\ &\rightarrow \mathbf{K}\,\mathrm{permit}(\mathrm{Alice}, ?\,\mathrm{sbj}, \mathrm{READ}, ?\,\mathrm{rsc}, \text{HIGH-PRIORITY})\end{aligned} \tag{6}$$

For knowledge representation, we usually assume our knowledge about the environment is not complete. Consequently, logics that are designed for knowledge representation such as description logic usually designed based on the open-world assumption. In contrast, for enforcing access control, we typically prefer to use the closed-world assumption. In the reasoning based on the open-world assumption, if the knowledge base does not infer α , we cannot assume $\mathbf{not}\,\alpha$. In other words, in such a reasoning, there are some predicates that we do not know whether they are true or false. In contrast, in the closed-world reasoning, we assume our knowledge about the environment is complete. Consequently, in the closed-world reasoning, if the knowledge base does not infer α , we conclude $\mathbf{not}\,\alpha$. For example, if a user does not establish a friendship relationship with Alice, according to the closed-world assumption, we assume the user is not a friend of Alice; however, in our knowledge base, we do not have any predicate indicating such a fact. The closed-world in comparison to the open-world assumption is a better assumption for enforcement of access control in OSNs, because it does not leave any access request without answer.

In the nonmonotonic reasoning, it is conceivable that the size of the inferred predicates is shrunk by adding new information to the knowledge base. In particular, for enforcing access control in OSNs, typically if a user establishes more relationships with others, he/she can access more resources. However, in some circumstances, establishing new relationships with a user can lessen resources accessible by the user. Thus, supporting nonmonotonicity is crucial for enforcing access control in OSNs. For example, according to Rule (7), users who are Alice's friends and are not Alice's colleagues are permitted to read her notes. Consequently, as soon as one of the Alice's friends establishes a colleague relationship with her, his/her request to access Alice’s notes will be denied. In fact, these kinds of rules cannot be expressed in the existing ontology based access control models [5], [6] proposed for OSNs, as they assume the open-world assumption for defining their rules.

$$\begin{aligned}&\mathbf{K}\,\text{o}(?\text{rsc}), \mathbf{K}\,\text{s}(?\text{sbj}), \mathbf{K}\,\text{IS-FRIEND-OF}(\text{Alice}, ?\text{sbj}),\\ &\mathbf{not}\,\text{IS-COLLEAGUE-OF}(\text{Alice}, ?\text{sbj}), \mathbf{K}\,\text{NOTE}(?\text{rsc})\\ &\rightarrow \mathbf{K}\,\text{permit}(\text{Alice}, ?\text{sbj}, \text{READ}, ?\text{rsc}, \text{HIGH-PRIORITY})\end{aligned} \tag{7}$$

B. Conflict Resolution and Access Control Decision

Users can define access control rules for protecting their resources. In some conditions, these access control rules can be in conflict with each other. Conflict between two rules occurs when one of these rules prohibits, while the other one permits an action on a resource. These conflicts can be categorized to *permanent* and *potential* conflicts. In permanent conflicts, the antecedents of two rules are equivalent but their consequents are contradictory. For example, suppose Alice has defined an access control rule which permits her colleagues to read her notes. Then, Alice decides to define a new rule which prohibits her colleagues to read her notes. These conflicts usually occur when a user makes a mistake in the definition of access control rules and should be eliminated. Such conflicts are supposed to be identified when a new access control rule is being inserted into the knowledge base. As a resolution method, more priority can be considered for either the more recent or the older access control rules.

Some conflicts can occur on special conditions. These kinds of conflicts differ from permanent conflicts mentioned above. In fact, these rules do not conflict with each other in all states. In our proposed model, various priority levels can be assigned to rules for resolving such conflicts. Two approaches can be supposed for defining priority levels for rules:

1) The OSN provider can define a set of priority levels as well as their inter-relationships. Hence, all the users must use the same set of priority levels in this case.
2) Each user can define his/her set of priority levels and their inter-relationships.

The relationships between priority levels can be either *total* or *partial* order. To provide more flexibility, we suppose that priority levels can be defined by each user and partial order relationships between such levels are supported in the model. In order to consider priority levels defined by users in the reasoning procedure, several predicates are defined. These predicates are presented in TABLE II. By using “*hasMorePriority”* predicate*,* users can define the relationship between two priority levels. The first parameter of this predicate represents the user who defines the priority levels and the two other parameters represent the two comparing priority levels. In fact, user assigns higher priority to the first priority level. Moreover, *h-permit* (*h-prohibit*) will be inferred for the specific priority level from the knowledge base provided that permission (prohibition) with either higher or incomparable priority level is inferred. In the last step of the access control procedure, *fd-permit* or *fd-prohibit* will be inferred. In fact, these predicates determine whether the request should be granted or denied from the view point of the authority.

Rule (8) presented in TABLE III is defined to enforce transitivity relations among priority levels defined by a user. Additionally, two scenarios are conceivable for comparison of the priority levels assigned to the contradictory access control rules:

1) The two user-defined priority levels are comparable and one of them has more priority. In this case, Rule (9) and Rule (10) presented in TABLE III can be used to demonstrate the existence of the privilege with the higher priority level in the knowledge base.
2) The two priority levels are incompatible or having the same priority levels. In this case, a user may consider more priority to either the permission or the prohibition. If a denial-takes-precedence policy is chosen, Rule (11) can be used for giving more priority to the negative privileges than positive privileges. Otherwise, Rule (12) can be used.

If *permit* (*prohibit*) is inferred for a specific priority level and *h-prohibit* (*h-permit*) is not inferred for that level, then *fd-permit* (*fd-prohibit*) will be inferred. Therefore, Rule (13) and Rule (14) can be used for conflict resolution among users’ access control rules with various priority levels. In some conditions, it is possible to infer neither *fd-permit* nor *fd-prohibit* for a specific request from the access control rules

TABLE III. Different Types of Rules in Proposed Model

Type	Rules
Transitivity property	$\mathbf{K}\ \text{hasMorePriority}(?a, ?p_1, ?p_2), \mathbf{K}\ \text{hasMorePriority}(?a, ?p_2, ?p_3) \rightarrow \mathbf{K}\ \text{hasMorePriority}(?a, ?p_1, ?p_3)$ (8)
Propagation of privileges in the priority levels	$\mathbf{K}\ \text{permit}(?a, ?sbj, ?act, ?rsc, ?p_1), \mathbf{K}\ \text{hasMorePriority}(?a, ?p_1, ?p_2) \rightarrow \mathbf{K}\ \text{h-permit}(?a, ?sbj, ?act, ?rsc, ?p_2)$ (9) $\mathbf{K}\ \text{prohibit}(?a, ?sbj, ?act, ?rsc, ?p_1), \mathbf{K}\ \text{hasMorePriority}(?a, ?p_1, ?p_2) \rightarrow \mathbf{K}\ \text{h-prohibit}(?a, ?sbj, ?act, ?rsc, ?p_2)$ (10)
Denial-takes-precedence	$\mathbf{K}\ \text{prohibit}(?a, ?sbj, ?act, ?rsc, ?p_1), \mathbf{K}\ \text{permit}(?a, ?sbj, ?act, ?rsc, ?p_2), \textbf{not}\ \text{hasMorePriority}(?a, ?p_2, ?p_1),$ $\textbf{not}\ \text{hasMorePriority}(?a, ?p_1, ?p_2) \rightarrow \mathbf{K}\ \text{h-prohibit}(?a, ?sbj, ?act, ?rsc, ?p_2)$ (11)
Permit-takes-precedence	$\mathbf{K}\ \text{prohibit}(?a, ?sbj, ?act, ?rsc, ?p_1), \mathbf{K}\ \text{permit}(?a, ?sbj, ?act, ?rsc, ?p_2), \textbf{not}\ \text{hasMorePriority}(?a, ?p_2, ?p_1),$ $\textbf{not}\ \text{hasMorePriority}(?a, ?p_1, ?p_2) \rightarrow \mathbf{K}\ \text{h-permit}(?a, ?sbj, ?act, ?rsc, ?p_2)$ (12)
Conflict resolution	$\mathbf{K}\ \text{permit}(?a, ?sbj, ?act, ?rsc, ?p), \textbf{not}\ \text{h-prohibit}(?a, ?sbj, ?act, ?rsc, ?p) \rightarrow \mathbf{K}\ \text{fd-permit}(?a, ?sbj, ?act, ?rsc)$ (13) $\mathbf{K}\ \text{prohibit}(?a, ?sbj, ?act, ?rsc, ?p), \textbf{not}\ \text{h-permit}(?a, ?sbj, ?act, ?rsc, ?p) \rightarrow \mathbf{K}\ \text{fd-prohibit}(?a, ?sbj, ?act, ?rsc)$ (14)
Default policy	$\textbf{not}\ \text{fd-prohibit}(?a, ?sbj, ?act, ?rsc), \textbf{not}\ \text{fd-permit}(?a, ?sbj, ?act, ?rsc) \rightarrow \mathbf{K}\ \text{fd-prohibit}(?a, ?sbj, ?act, ?rsc)$ (15)

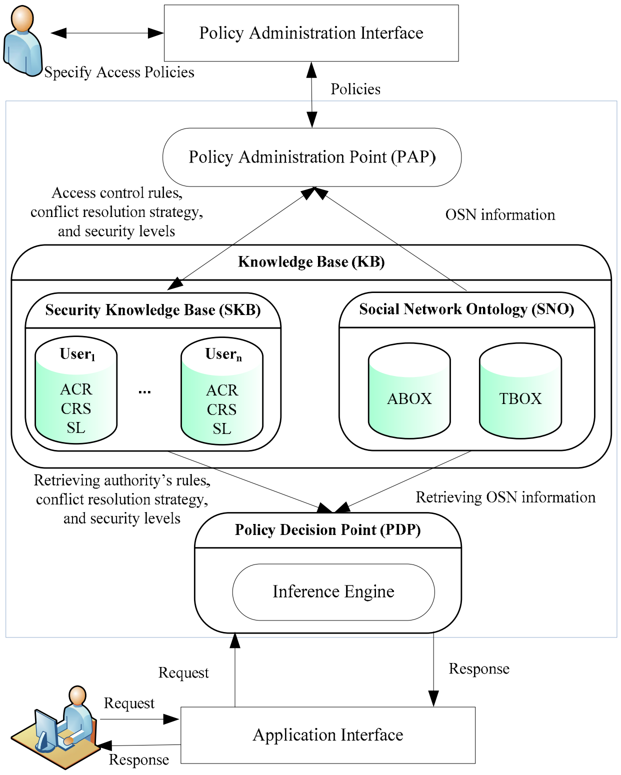

Specify Access Policies
Policy Administration Interface
Policies
Policy Administration Point (PAP)
Access control rules, conflict resolution strategy, and security levels
OSN information
Knowledge Base (KB)
Security Knowledge Base (SKB)
User₁
ACR
CRS
SL
...
Userₙ
ACR
CRS
SL
Social Network Ontology (SNO)
ABOX
TBOX
Retrieving authority's rules, conflict resolution strategy, and security levels
Retrieving OSN information
Policy Decision Point (PDP)
Inference Engine
Request
Response
Request
Response
Application Interface

Figure 2. The Proposed Architecture for Enforcing Access Control

defined by a user. To decide whether the system should grant or deny such request, the close policy as the default policy is defined. Therefore, if *fd-permit* or *fd-prohibit* for a request cannot be inferred, the request will be denied. Rule (15) is defined to enforce this policy. In fact, considering the open policy may cause privacy issues for users. Regardless of the fact that our model can support the open policy, we avoid defining the open policy as the default policy.

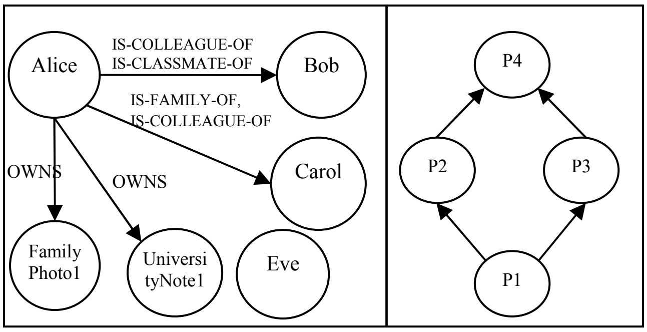
C. *Access Control Enforcement*

Architecture considered for enforcing proposed access control model is shown in the Fig. 2. An authority (owner) can use Policy Administration Interface (PAP) for defining his/her access control rules (ACR), conflict resolution strategy (CRS), and security levels (SL). Moreover, Application Interface is used for sending users' access requests to the system. An access request is modeled as a triple (subject, action, object). Knowledge base can be divided to the Security Knowledge Base (SKB) and the Social Network Ontology (SNO). SNO is divided to TBox, which contains concepts and their relationships, and ABox, which contains individuals. SKB contains access control policies defined by users. Amount of time taken for performing inference task is increased significantly if all the ACR defined by all the users are used in the reasoning procedure. To avoid this problem, users' access control policies are stored separately and ACR, CRS, and SL used in the reasoning procedure are restricted to the owner's. According to the defined access control policies and SNO, Policy Decision Point (PDP) is responsible to decide whether the request must be granted or denied. After receiving a request to access a resource, following steps are taken:

1) A query is sent to the knowledge base to discover the owner of the object.
2) The set of access control rules, the priority levels defined by the authority, and the OSN information are retrieved from the knowledge base.
3) If the predicate, *fd-permit*(*authority*, *subject*, *action*, object) is inferred from the knowledge base, the request is granted. Otherwise, the request is denied.

D. *Case Study*

For the sake of representing how our proposed access control model is applicable to OSNs, an imaginary OSN is modeled. Suppose the social graph and the security levels represented in Fig. 3. In this scenario, Alice would like to share her family photos and university notes in the OSN with some users. The set of access control rules defined by Alice to protect her resources, and various priority levels considered for her rules are shown in TABLE IV. According to Rule (16) and Rule (17), she permits her family members to see her family photos and prohibits her colleagues to access them. In addition, she would like to share her university notes with her classmates and not



(a) (b)

Figure 3. (a). An instance of OSN (b). Alice's priority levels

TABLE IV. Alice's access control policy rules

Rules	
K o(? rsc), **K** s(? sbj), **K** IS-FAMILY-OF(Alice, ? sbj), **K** FAMILY-PHOTO(? rsc) → **K** permit(Alice, ? sbj, READ, ? rsc, P4)	(16)
K o(? rsc), **K** s(? sbj), **K** IS-COLLEAGUE-OF(Alice, ? sbj), **K** FAMILY-PHOTO(? rsc) → **K** prohibit(Alice, ? sbj, READ, ? rsc, P3)	(17)
K o(? rsc), **K** s(? sbj), **K** IS-CLASSMATE-OF(Alice, ? sbj), **K** UNIVERSITY-NOTE(? rsc) → **K** permit(Alice, ? sbj, READ, ? rsc, P3)	(18)
K o(? rsc), **K** s(? sbj), **K** IS-COLLEAGUE-OF(Alice, ? sbj), **K** UNIVERSITY-NOTE(? rsc) → **K** prohbit(Alice, ? sbj, READ, ? rsc, P2)	(19)

TABLE V. Parts of Reasoning Results

Inferred Access Control Predicates
permit(Alice, Carol, READ, FamilyPhoto1, P4), prohibit(Alice, Carol, READ, FamilyPhoto1, P3), permit(Alice, Bob, READ, UniversityNote1, P3), prohibit(Alice, Bob, READ, UniversityNote1, P2)
fd-permit(Alice, Carol, READ, FamilyPhoto1), fd-prohibit(Alice, Bob, READ, UniversityNote1), fd-prohibit(Alice, Eve, READ, FamilyPhoto1), fd-prohibit(Alice, Eve, READ, UniversityNote1)

to share these notes with her colleagues. Rule (18) and Rule (19) are defined for this purpose, respectively. Moreover, Alice considers a denial-takes-precedence approach for conflict resolution among access control rules. According to the defined rules, conflicts among access control rules occur if a user such as Carol, who has both the family and colleague relationships with Alice, sends a request to access Alice's family photo. In this context, regarding the fact that Alice assigned higher priority to Rule (16) than Rule (17), the request will be permitted. In contrast to the rules defined for protecting family photos, the priority levels of rules defined for protecting university notes are incomparable to each other. Consequently, since higher priorities are assigned to the negative authorizations by Alice, the requests of users such as Bob for accessing Alice's university notes will be denied. Moreover, according to the default policy, requests of users such as Eve will be denied. Some important predicates inferred from the described access control rules are shown in TABLE V.

VI. Conclusions and Future Works

In this paper, a prioritized ontology based access control model for OSNs was proposed. Description logic, in this model, is used for modeling OSNs and logical rules are used for expressing access control rules. In fact, supporting nonmonotonic reasoning is essential for enforcing access control in OSNs. Consequently MKNF^{+} is chosen as the formalism for integration of DL and rules. In addition, approaches for enforcing these rules by MKNF^{+} are mentioned and various conflicts among access control rules defined by a user are analyzed and different methods such as defining priorities, proposed for resolving these conflict.

In the proposed model, users assign priorities of rules themselves. If priorities of rules are labeled automatically according to the predicates appear in the antecedents of rules, the complexity of defining access control rules by the users will be diminished significantly. Therefore, proposing an automatic mechanism for determining priority of rules is left as a future work. Furthermore, defining access control policies in the MKNF^{+} format has some difficulties for average users. Thus, designing user-friendly interfaces and converting users' access control policies to MKNF^{+} rules, can be mentioned as an important future work.

References

[1] B. Motik, "Reconciling Description Logics and Rules," Journal of the ACM (JACM), vol. 54, no. June, pp. 1-63, 2010.

[2] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," Journal of Computer-Mediated Communication, vol. 13, no. 1, pp. 210–230, 2007.

[3] B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks," ACM Transactions on Information and System Security, vol. 13, no. 1, pp. 1–38, 2009.

[4] P. W. L. Fong, M. Anwar, and Z. Zhao, "A Privacy Preservation Model for Facebook-Like Social Network Systems," in Proceedings of the 14th European conference on Research in computer security, 2009, vol. 659, no. April, pp. 0-18.

[5] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "A semantic web based framework for social network access control," in Proceedings of the 14th ACM symposium on Access control models and technologies, 2009, pp. 177–186.

[6] A. Masoumzadeh and J. Joshi, "Osnac: An ontology-based access control model for social networking systems," in IEEE International Conference on Privacy, Security, Risk and Trust, 2010, pp. 751–759.

[7] E. Bertino, S. Jajodia, and P. Samarati, "Supporting multiple access control policies in database systems," in IEEE Symposium on Security and Privacy, 1996, no. 1, pp. 94–107.

[8] F. Cuppens, N. Cuppens-Boulahia, and A. Miege, "Inheritance hierarchies in the Or-BAC Model and application in a network environment," in Second Foundations of Computer Security Workshop(FCS04), 2004, pp. 41–60.

[9] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, "AL-log˜ : Integrating Datalog and Description Logics," Journal of Intelligent Information Systems, vol. 10, no. 3, pp. 227-252, 1998.

[10] R. Rosati, "DL + log˜ : Tight Integration of Description Logics and Disjunctive Datalog," in The Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR2006), 2006, pp. 68-78.

[11] A. Y. Levy and M. C. Rousset, "CARIN˜ : A Representation Language Combining Horn rules and Description Logics," in Proceedings of the 12th European Conf. on Artificial Intelligence (ECAI-96), 1996, pp. 1-50.

[12] F. M. Donini, D. Nardi, and R. Rosati, "Description logics of minimal knowledge and negation as failure," ACM Transactions on Computational Logic (TOCL), vol. 3, no. 2, pp. 177–225, 2002.