A Framework for Privacy in Social Communities

Mohammad M. R. Chowdhury*
Najeeb Elahi
Sarfraz Alam and
Josef Noll

UniK-University Graduate Center,
Post Box 70, N-2027 Kjeller, Norway
Fax: +47 63 81 81 46 E-mail: {mohammad, najeeb, sarfraz, josef}@unik.no
*Corresponding author

Abstract: Web based social communities are one of the most widely used applications nowadays. Recently, privacy concerns within these communities have increased significantly. This paper proposes a framework which addresses these challenges by introducing a distributed social community mechanism and relation based content access management by exploiting Semantic Web technologies. In this regard, a community framework along with fine-grained access mechanisms are formalised using the Web Ontology Language. Instead of an explicit definition, some additional facts of the mechanisms are inferred by executing access authorisation policies through Semantic Web rules. These implicit facts answer the queries of the information requesters, and optionally can be passed back to the formalised knowledge base to check the validity and consistency. This paper also presents the components of functional architecture of the framework.

Keywords: privacy; social community; ontology; distributed mechanism; access management.

Biographical Notes: Mohammad M. R. Chowdhury is PhD candidate at the University Graduate Center at Kjeller (UniK), Norway in the area of User Mobility and Service Continuity. He received his MSc from Helsinki University of Technology in Radio Communication. His current research interests are identity management and identity based service interactions, seamless user experience in heterogeneous wireless networks.

Najeeb Elahi is Scientific Assistant at the University Graduate Center at Kjeller (UniK), Norway in the area of Social Community. He worked in Digital Enterprise Research Institute (DERI) Ireland and his current research interest focuses on Semantic Web technologies and applications.

Sarfraz Alam is PhD candidate at UniK, University Graduate Center in Kjeller, Norway. His research area covers Mobile SOA, Semantic Mobile Services and SOA security and privacy. He received his M.Sc. degree in the area of Information Security from The Royal Institute of Technology (KTH), Sweden.

Josef Noll is Professor at the University Graduate Center at Kjeller, Norway (UniK) in the area of Mobile Systems. He is also Senior Advisor in Movation. He received his Ph. D. from University of Bochum (D), worked for European Space Agency at ESTEC from 1991-1997, and from 1997-2005 at Telenor R&D. His working areas include mobile authentication, wireless broadband access, personalized services, mobile-fixed integration, and the evolution to 4G system.

1 Introduction

It is increasingly important to bring people together and to promote mutual understanding, learning and knowledge exchange in social and professional communities. Creating *virtual or online communities* through social networking and content-sharing sites are one of the major ways to do this. Though one of the objectives of these communities was to help us work together over common activities or interests, currently users are more interested to boost their friends list only. To establish social connections meaningful within such communities, we are proposing a framework of a community platform that can facilitate sharing of private data or contents with only the concerned people in the communities besides the customary public contents sharing within the community boundary. Contrary to the current online community architecture, we are suggesting a decentralised mechanism where individual profiles are located at different places and the owners of the profiles control the disclosure of information to the community or its members. In this paper, we are going to focus on the formal representation of the framework containing the relation based content access management and the distributed community mechanism. These two features have considerable privacy implications which will be addressed in detail in chapter 2.1. Moreover, we present the overall framework and architecture and implement some of the core concepts.

The paper is organised as follows, the next chapter describes our motivation and illustrates a use case scenario. Chapter 3 discusses the key features of the proposed framework and the technologies used to design it. Chapter 4 briefly introduces the components of the functional architecture. The key focus areas of this work, the community knowledge base representation and policy enforcement engine specification are depicted in detail in chapter 5 and 6. The next chapter includes the evaluation and related work. The paper ends with a conclusion and a brief discussion on future work.

2 Motivation

This chapter discusses the shortcomings of privacy handling in online communities and illustrates through a use case scenario how the contents access constraints can be handled.

2.1 Problem statement

Nowadays social networks are one of the most widely used platform of online communities. According to the report from ComScore (2007) the famous social network sites like MySpace, Facebook, LinkedIn, Friendster, Orkut, Bebo, etc., are enjoying about 65 million daily visitors and the growth rate is 50% to 300%. People joining several sites need to re-declare their friends and re-publish their contents on every site they register. It is time consuming and also distracts them from joining new communities. Besides, these actions pile up a huge amount of redundant and incomplete information on the Web. Breslin (2007) pointed out these shortcoming and proposed reusable profiles to let the communities import existing connections and contents from other storages. Our concern is more that centralised architectures compromise the member's privacy. Brad Fitzpatrick (2007), the founder of the Live Journal blogging community, wrote in an article the importance of forming decentralised social relations. Therefore, decentralisation and reuse of user profiles are very crucial from both a privacy and a maintenance point of view.

Privacy concerns within these online social communities have increased recently. Initially access to resources (photos, videos, personal details) in most web based community sites were open. Gradually privacy features have been integrated into these sites through the "friends" attribute but with no further granularity. A binary decision on "friend" or "not friend" is not an appropriate representation of the social life and an issue for privacy enhancement. Nowadays, we increasingly find the presence of our family members, relatives, neighbours, the best friends, not so intimate friends in the same online communities. But in most cases they are all categorised under the same "friends". With the increasing variations of intimacy in the relationships, we expect customised divisions in this privacy feature. Currently, online communities are dealing with access to public data or contents.

Our proposed framework also includes private content access, where privacy enhancement mechanisms will be a prerequisite. The proposed mechanism is focusing on the granularity of privacy features and consists of a decentralised architecture which is expected to mitigate these problems.

2.2 Use case scenario

Figure 1 shows our use case scenario where we provide an example of a community platform and a personal profile of an individual who joins as a member of one of the communities. Personal profiles are not centrally maintained within the community platform. They are distributed and owners have full control on which information or content will be disclosed to the community. Afterward, a fine-grained content access mechanism ensures that only people with right relations are allowed to access the resources. Members can even share their private resources with only specific people. It is assumed here that every person belongs to at least one community. Our example is illustrated in figure 1 and consists of a community platform with two communities: Cycling and Rowing. Josef, Sarfraz, and Mushfiq are members of the Cycling community, and George and Najeeb belong to the Rowing community. Private resources of the community correspond to the private resource of a member. Public resources are further divided into two more resource groups. Public to community resources can be accessed only by the same com-

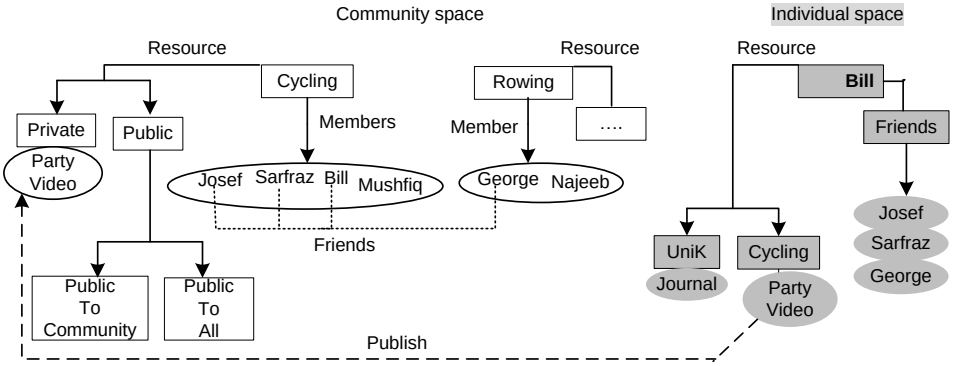

Figure 1 A community platform use case scenario.

munity members and public to all resources are visible to all. Bill recently joins the Cycling community. He used to store his resources (according to a hierarchy) and maintains a list of friends in his profile. Besides, he considers Josef as a more intimate friend than Sarfraz. Bill wants to share a private party video only to specific members of the Cycling community. These specific members are related to the community, but they are also very close friends of Bill. Access rights to these private and community resources need to meet various constraints. These are described as follows:

- Fellow community members who are also intimate friends of the resource owner can get full access to his private resources. Here, Josef gets full access to Bill's private resource, a party video.
- Fellow community members who are not an intimate friends of the resource owner's will get limited access to his private resources. In our example, Sarfraz gets limited access to Bill's party video.
- When the resource requester is an intimate friend of its owner but belongs to different community, then the requester gets limited access to owner's private resources. This is true for George trying to access Bill's party video.
- When the requester is neither a friend of the resource owner nor belongs to the same community, such access request will be denied. In our example Najeeb cannot get access to Bill's party video.
- The members of the same community will get full access to the community resources. In our example Josef, Bill, Mushfiq, and Sarfraz will get the full access.
- When the requester and the resource owner belong to different communities, then they will get limited access to each other's community resources.

The limited access allows the information requesters only to view the descriptions of the resources whereas the full access provides full multimedia access to the resources.

3 Privacy framework

The proposed framework is intended to provide privacy in virtual communities. A relation based content access management and distributed community mechanism are the key features of it. This chapter introduces the key components of the framework along with the core technologies used to design it.

3.1 Relation based content access management

We establish the community structure based on the social structure or the social network (Abercrombie et al., 2000). The structure consists of nodes (generally individuals or organisations) that have edges representing one or more specific types of relations, such as possessing contents, having friends & families etc. Such structure is represented as graphs which model relationships between things in mathematics. Later, Hanneman (2005) suggested the introduction of graphs to represent social relations. In figure 2, we bring in the concepts of Social Graph (SoG) and Social Content Graph (SoCG) which delineate individual's and content's social relations with the social community. These graphs are the basis of our community knowledge base representation.

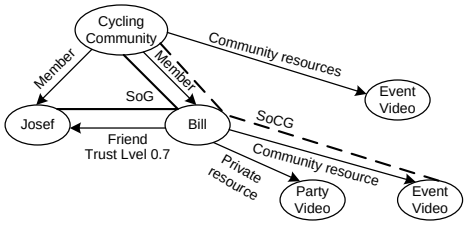

Figure 2 The Social Graph (SoG) and Social Content Graph (SoCG).

Managing content access with considerable granularity of constraints is not a trivial job. These constraints are formulated considering various levels of relationships with the individuals and the communities. In the proposed framework, we include trust levels along with friend relations to indicate the closeness of friendship. Special intimacy deserves a privilege. So, we consider it as one of the constraints to access somebody's private resources. Being member of the same community suggests sharing of similar interests.

Our work takes into account multiple combinations of these two aspects of relations to formulate the content access restrictions. The use of Semantic Web Technologies allows us to devise such fine-grained access constraints which enhance the privacy of social communities.

3.2 Distributed social community mechanism

The proposed virtual social community is based on a decentralised architecture. Each individual is assumed to maintain his own profiles at dedicated spaces where he declares his friends, the communities he belongs to, and uploads his contents. Each time the person joins a new community, he discloses the appropriate friends or contents to the community through a mapping service. It is assumed that the

community platform provides this mapping service. The proposed framework not only makes the system more manageable, it also avoids redundancy as content is only located once in the network. Maintaining decentralised profiles is found to be crucial from a privacy point of view. The user might decide to store data in a private place or ask a trusted third party to provide a profile storage facility.

3.3 Generic functional architecture

This section provides a generic functional architecture (see figure 3) with a brief introduction of each component of the framework. The functional architecture includes the community interface, a Semantic Query interface (SPARQL End Point), a Semantic Policy Enforcement Engine (SPEE) and a Social Semantic Knowledge Base.

The community interface facilitates the web based access through a web interface, includes user authentication, and handling of queries toward the SPARQL End Point It further includes a knowledge base management tool.

A SPARQL endpoint is a conformant SPARQL protocol service which enables users to query a community knowledge base using the SPARQL query language. The endpoint returns the query results in machine processable formats.

The SPEE enforces the access authorisation policies, and the Knowledge Base stores the semantic descriptions of the community and individual profile structures and relations. In this proposed framework we are focusing only on the design of the Social Semantic Knowledge Base and the SPEE. For the clarity, we include a brief description of the remaining components in chapter 4.

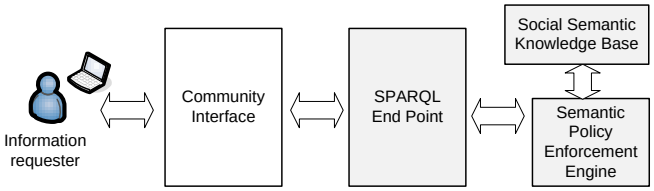

Figure 3 Generic functional architecture for semantic based access rights handling

3.4 Core technologies

This section introduces the basics of the core technologies we use in this work. Our implementation based on Semantic Web Technologies ensures the required fine-grained access to content in a community platform.

3.4.1 Semantic Web technology

Semantic Web is seen as the next generation of information management systems on the Web. Berners-Lee (2001) first envisioned the promise of Semantic Web. It provides various technologies to formally represent a domain knowledge to facilitate the understanding and manipulation by computers. Ontologies (Fensel, 2004) are the cornerstone technology of Semantic Web. It is used to capture the

knowledge about a domain of interest in the form of concepts and their relationships. Semantically rich representations (through ontology) permits description of community structures, resources, members, and constrained access situations at different levels of abstraction and support reasoning about both the structures and the properties of the elements that constitute the system.

Among the different ontology languages, we are focusing on the Web Ontology Language (OWL) which is suggested by the World Wide Web Consortium (W3C) (Smith et al., 2004). OWL builds on RDF (Klyne et al., 2004) and RDF Schema (RDFS). OWL is chosen because it facilitates greater machine interpretability of the Web content than that supported by XML, RDF, and RDFS by providing additional vocabulary along with a formal semantics. There are three species of OWL: OWL Lite, OWL DL and OWL Full and these are designed to be layered according to their increasing expressiveness.

3.4.2 Enhance expressibility

The framework is supposed to evaluate the relations and grant permissions to access resources. Therefore, it seems necessary to enhance the expressibility of the ontology knowledge by adding reasoning support. We decided to use OWL DL which is based on Description Logics (hence the suffix DL). These are the decidable part of First Order Logic[1] and are therefore amenable to automated reasoning. Though OWL DL lacks in expressivity power compared with OWL Full, it maintains decidability[2] and regains computational efficiency. The computational efficiency is an important feature since the scheme has to handle scores of social relations.

As the expressivity provided by the OWL is limited by tree like structures (Motik et al., 2004), the implicit knowledge cannot be inferred from the indirect relations between entities. However, the proposed framework spends most of its power to infer such relations that will determine the outcome of our restricted access scenarios. To be able to infer these knowledge, rule support and interworking with ontologies must be taken into account. One suitable rule language is the Semantic Web Rule Language, SWRL (Horrocks et al., 2004), supporting complimentary features for OWL DL. SWRL rules are written in terms of OWL classes, properties and individuals, and are defined as a set of antecedent and consequent parts. For example, a SWRL rule expressing that a person with a male sibling has a brother looks like as follows,
$Person(?p) \wedge hasSibling(?p, ?s) \wedge Man(?s) \longrightarrow hasBrother(?p, ?s)$
It would require capturing the concepts: person, male and properties: hasSibling and hasBrother from OWL and p and s are variables.

4 Component descriptions of functional architecture

This chapter briefly introduces the components of the proposed privacy framework. Main focus will be on the Social Semantic Knowledge Base and Semantic Policy Enforcement Engine as being the critical components of the framework.

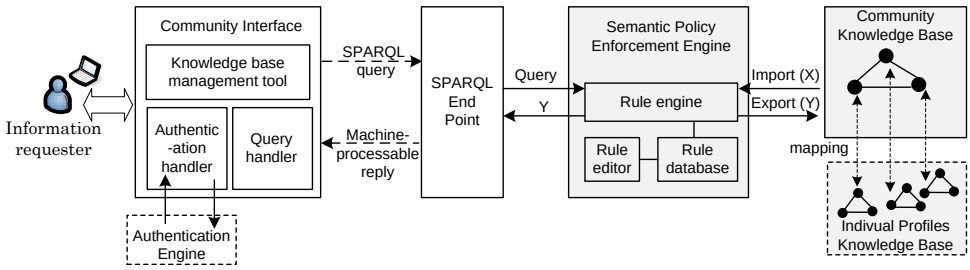

Figure 4 Detailed functional architecture of the proposed social community framework.

4.1 Community interface

The community interface consists of three components. Each component works as a proxy on behalf of the information requester for a specific task.

Authentication handler There are flexibilities of using different authentication methods ranging from simple user name/password to SIM-based authentication. Upon receiving an authentication request, the authentication handler consigns the request to the authentication engine which performs the authentication and send back the response to the originator. Chowdhury (2007) proposed an integrated identity mechanism consists of certificates, keys and preferences. This mechanism can be integrated with the proposed community interface by generating and distributing community keys to the members' devices and thereby authenticate with these keys.

Query handler It manipulated SPARQL queries and acts as a delegator for the information requester. It formulates the outbound query expression based on user attributes and also interprets the query result.

Knowledge base management tool It provides the facility of ontology creation, administration and management. Knowledge management can be performed locally and remotely over Internet protocols like http(s).

4.2 SPARQL end point

The Knowledge Base stores the ontologies as RDF triples. SPARQL (Prudhommeaux, 2008) emerges as a standard query language for the RDF triples. It can be used to express queries across diverse sources, whether the data is stored natively as RDF or mapped into RDF from customary RDBS. Lu (2007) in a paper stored the ontology in RDBS and used translator/adapter to use SPARQL query language over it. A SPARQL end point is the de facto standard to expose large data stores on the Semantic Web. It conforms to the SPARQL query protocols. In the proposed framework, the knowledge base is linked with the SPARQL end point through an interface and the RDF triples of the knowledge base can be access only by sending the SPARQL queries to the end point. It then returns the query results in machine processable formats. The endpoint communicates with the SPEE for query resolution.

4.3 Semantic policy enforcement engine

Semantic Policy Enforcement Engine (SPEE) provides the authorisation decisions according to the authorisation policy definitions which answers the facts that who can access to what resources with what privileges (full access, limited access and access denied). It consists of rule engine, rule database and rule editor. There is a SWRL rule editor where Members are allowed to edit own rules for customisation of privacy support. SPEE is using the Jess rule engine and takes the SWRL rules from Rule database. The rule engine executes the rules and the inferred facts are supplied back to the SPARQL end point in response to the queries. These can even be exported back to the OWL knowledge base for checking the *validity* of the policies and the *consistencies* of the designed knowledge base. SPEE is discussed more in detail in chapter 6.

4.4 Social semantic knowledge base

The Social Semantic Knowledge Base contains the formal representation of the community ontologies in OWL. In practice, the Knowledge Base is a RDF triple database. OWL ontologies are commonly serialised using RDF/XML syntax and can be easily mapped to the RDF triples. Each triple consists of a subject, predicate and an object and represents a statement of a relationship between the two things. The concept is similar to the graphs described in chapter 3.1. The community ontology is created from the individual ontologies through mapping. These mappings are done by the owners of the individual ontologies through the knowledge base management tool. Individual ontologies can be located in data bases on the client side or managed by trusted third parties.

5 Ontology describing community and access management semantics

The Social Semantic Knowledge Base is maintained in the form of ontologies and we use OWL DL to formally represent a social community platform and an individual profile. To avoid complexity, we limit the scope of individual profiles to the description of friends and contents. This chapter also includes the mapping between the components of community ontology and individual profile ontology which is a basis for distributed ontology mechanism. The Protege ontology editor platform (Protege 3.4) is used to design these ontologies and it contains a separate individual ontology (*BillOnt.*) and a community ontology (*CommunityOnt.*). For the detail descriptions of ontologies we use the *Description Logic* syntax.

5.1 Preliminaries

Through ontologies, we formally represent the notion of Social Graphs and Social Content Graphs (figure 2) according to the use case scenario described in chapter 2.2. Here, the nodes (*Instances*) of similar concept are grouped under *Classes* so that they inherit the same relations (*Properties*) or restrictions. The formalisation of knowledge facilitates the inference of implicit knowledge using the SPEE (addressed in chapter 6). An ontology is a set of *classes* C, *properties* P and *instances*

i. In this work, concepts have three types of relation among them: *subClassOf*, *disjointWith*, and *equivalentClass*. The semantic scope (SC) of a concept (class) C_i is represented as $(SC\,(C_i))$. The definition of these three types of relations are,

- *subClassOf*: $SC\,(C_1) \subseteq SC\,(C_2)$, the semantic scope of C_1 is narrower than that of C_2.
- *disjointWith*: $SC\,(C_1) \subseteq \neg\, SC\,(C_2)$, the semantic scope of C_1 is disjoint with that of C_2.
- *equivalentClass*: $SC\,(C_1) \equiv SC\,(C_2)$, the semantic scope of C_1 is equivalent with that of C_2.

Here we also define several additional characteristics of OWL which we use here,

- $P\,(i_1, i_2)$ states that i_1 relates with i_2 through the property P.
- *owl:equivalentProperty*: $P_1 \equiv P_2$, it can be used to state that two properties have the same property extension.
- *owl:sameAs*: $i_1 \equiv i_2$, it is used here to state that two instances are in fact same.
- *owl:symmetricProperty*: it states that if P relates i_1 & i_2, then P also relates i_2 & i_1 and can be represented as $P \equiv P^-$, where P^- is the inverse property of P.

5.2 Defining concepts through classes

In the ontologies, the key concepts of the proposed domain (online social communities) are defined through *classes*. The individual ontologies contain owner, friends and resources. Resources are further subdivided into several classes. The class-subclass relations of the resources of the Bill ontology are as follows,

- $PrivateResource \subseteq Resource$
- $\{PublicToAllResource, PublicToCommunityResource\} \subseteq PublicResource \subseteq Resource$

In addition there are classes: *Owner* and *Friend* that contain no subclass relation. The community ontology contains the concepts: community, member and resources, and resources have subclasses addressing the user's content. Therefore, the community ontology (*CommunityOnt.*) contains the following class-subclass relations,

- $PrivateResource \subseteq Resource$
- $\{PublicToAllResource, PublicToCommunityResource\} \subseteq PublicResource \subseteq Resource$
- $Community \subseteq \neg\, Member \subseteq \neg\, Resource$, where $Community$ and $Member$ are defined as class. $Community$ has no subclass relation and $Member$ gets the subclass relation upon mapping (more in chapter 5.5).

5.3 Defining relations through properties

In the ontologies, a *property* relates two instances of the classes. Table 1 lists the properties used in the Bill and community ontology and the corresponding domains and ranges. Properties have a *domain* and *range*. Syntactically, *domain* links a property to a class and *range* links a property to either a class or a data range (Smith et al., 2004). From an instance point of view, a property relates instances from the *domain* with the instances from the *range*. Here *hasFriend* and *notFriendOf* properties are defined as *owl:symmetricProperty*. It ensures that when one person is added as a friend of another, then the opposite is also true. All the property relations are not explicitly defined, several relations are mapped (see the chapter 5.5)) and the relations, which provide the decisions to access appropriate resources, are defined from the facts derived through the inference in SPEE (see detail in chapter 6). These (the later) properties are: *hasFullAccess*, *hasLimitedAccess*, and *hasNoAccess*.

Table 1 The list of properties, their domains and ranges.

Ontology	Property name	Domain	Range
BillOnt	hasFriend	Owner	Friend
	hasTrustLevel	Friend	$\{0.1, 0.2....1.0\}$
	hasPrivateResource	Owner	PrivateResource
	hasPublicResource	Owner	PublicToAllResource
	hasCommunityResource	Owner	PublicToCommunityResource
CommunityOnt	hasMember	Member	Member
	hasFriend	Member	Member
	hasTrustLevel	BillOnt:Owner	$\{0.1, 0.2....1.0\}$
	notFriendOf	Member	Member
	hasPrivateResource	Member	PrivateResource
	hasPublicResource	Member	PublicToAllResource
	hasCommunityResource	Community	PublicToCommunity
	hasFullAccess	Member	Resource
	hasLimitedAccess	Member	Resource
	hasNoAccess	Member	Resource

To determine the closeness among the friends, the concept of trust is included through trust metric as described by Massa (2007). Trust metric is a measure of how an individual trusts his friends. In this work, trust metrics are values in percentage with range from 0.1 to 1.0 (with 0.1 interval) and they are hard coded by the profile owner (e.g. by Bill at BillOnt) against *hasTrustLevel* property. Bill assigns trust metric value 0.7 to Josef, 0.8 to George and 0.5 to Sarfraz. The trust metric value will be used later to measure the intimacy with friends and allow only the closest ones to access private resources (in chapter 6). The same property is also maintained in the community ontology. These two are mapped, so the later one can use the same values and restrictions imposed by the former (*BillOnt:hasTrustLevel*). Figure 5 illustrates all the classes and properties of the community and individual ontologies.

5.4 Realising concepts through instances

The concepts and relations between the concepts are defined through classes and properties. The real actors of a practical use case scenario are defined through

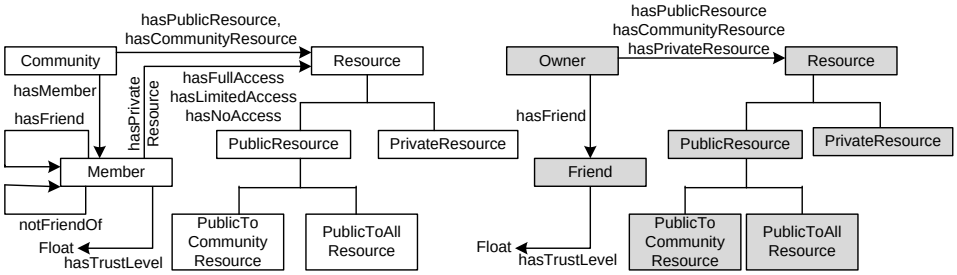

Figure 5 The classes and the properties of community and individual ontologies.

instances and they belong to the classes. The following are the instances of the Bill ontology,

- $\{George, Josef, Sarfraz\}$: Friend.
- *Bill*: Owner; $PrivatePartyVideo1$: PrivateResource; $HowToCycleVideo1$: PublicToAllResource; $CyclingPartyVideo$: PublicToCommunityResource.

Here we explicitly define who is the friend of whom and what are the resources the owner possesses and later we use mapping technique to make them available to the community ontology. The community ontology contains the following instances,

- $\{Josef, Sarfraz, Mushfiq, George, Najeeb\}$: Member; Bill is added as Member only when he is going to be mapped to the community ontology.
- $\{Cycling, Rowing\}$: Community.

Here we clearly define who are the members of which community. Josef, Bill (upon mapping), Sarfraz, Mushfiq are defined as member of Cycling community. George and Najeeb belong to the Rowing community. The answers to 'who has access (specific type of access) to which resource' are inferred when rules are executed in SPEE.

5.5 Mapping to support distributed ontology

To support distributed community mechanism, we need mediation between the components of the ontologies. Among the three widely used mediation techniques (Bruijn et al., 2006), we use ontology mapping which represents the correspondence between the ontologies. We have mapped few classes, properties and instances of the Bill ontology with that of community ontology. The following classes are mapped using *subClassOf* relation:

$BillOnt : Owner \subseteq Member$; $BillOnt : PrivateResource \subseteq PrivateResource$;
$BillOnt : PublicToAllResource \subseteq PublicToAllResource$;
$BillOnt : PublicToCommunityResource \subseteq PublicToCommunityResource$

The instances of the classes of the Bill ontology will be available to the community ontology only when they are mapped. It is assumed that the community

administrator will map the classes: Owner and Member. The remaining classes will be mapped by the owner of the individual ontology himself. Therefore, the resources will be disclosed to the community with the owner's consent. This is one of the features of the proposed ontology to support privacy.

The framework also maps some properties to avoid the explicit specification of some relations into the community ontology again. *hasFriend*, *hasTrustLevel* and *hasPrivateResource* properties of the Bill ontology are mapped with the same properties into the community ontology using *owl:equivalentProperty*. As a result, whenever the relation is specified in the Bill ontology, it will also be available to the community ontology and thereby supports the inference by the rules in SPEE. The mappings are represented as follows,

$BillOnt : hasFriend \equiv hasFriend$; $BillOnt : hasTrustLevel \equiv hasTrustLevel$;
$BillOnt : hasPrivateResource \equiv hasPrivateResource$.

But the community administrator has to explicity specify the relations with properties: *hasCommunityResource* and *hasPublicResource*. To support *hasFriend* mapping we also have to declare who are the persons common in both the ontologies (Bill & Community). Through mapping we realise the instances of the classes: Friend and Member who are in fact the same people. We use *owl:sameAs* and represented the relation as,

$BillOnt : Josef \equiv Josef; BillOnt : George \equiv George$;
$BillOnt : Sarfraz \equiv Sarfraz$.

Figure 6 illustrates the class browser of the community ontology in the Protégé platform. Apart from its own class-subclass hierarchy, it shows few classes of the *BillOnt*) as they are mapped to it.

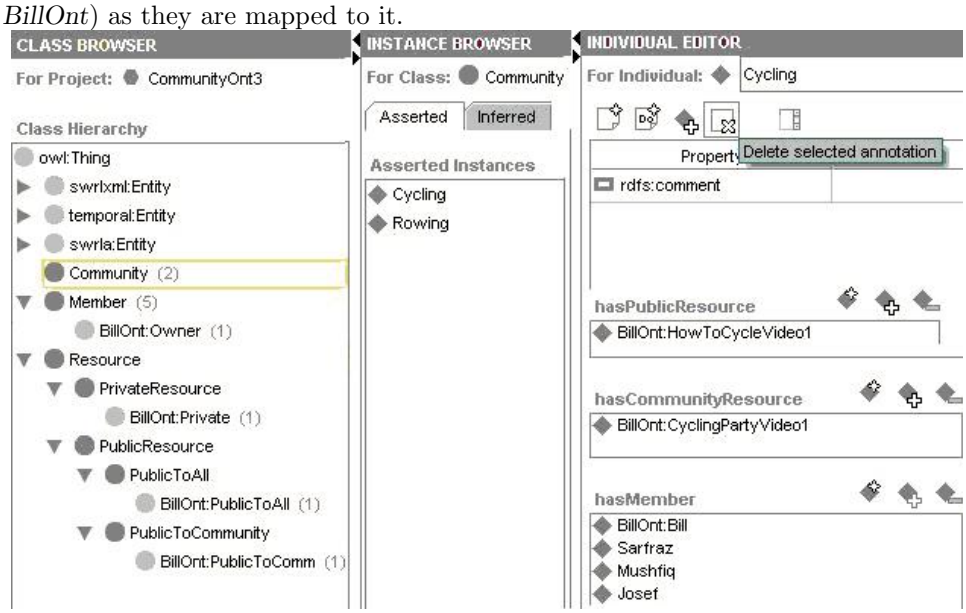

Figure 6 The class browser showing class-subclass hierarchies, few instances and relations.

6 Semantic policy enforcement engine

The SPEE contains the SWRL rule editor, the rule database and the Jess rule engine. The rule database is in fact the knowledge base itself and the Jess rule engine executes the SWRL rules. Besides generating authorisation policy decisions, SPEE also checks the consistency and validity of the policies. If X is the input from Community Ontology and Y is the decision derived from SPEE (see figure 4), relationships between these two can be described as, $Y = F(X) = F(P, X)$ where P is the ontology property, and $X \in \{C, i\}$, C and i are the ontology classes and instances respectively. F is the Jess rule engine execution environment for SWRL rules.

6.1 Authorisation policy definitions

We propose fine-grained authorisation policy definitions based on the multiple relations between the communities and their members. We assume that the framework is a community platform where every member belongs to certain communities.

Definition 1. Full access to private resources of individuals. M_1 gets full access to the private resources of M_2 when both belong to the same community and M_1 has trust level greater or equal to 0.7.

- $A \in SC\,(Community)\,;\ M_1, M_2 \in SC\,(Members)$
- $P_1, P_2, P_3 \in P$ where $P_1 = hasMember,\ P_2 = hasFriend,\ P_3 = hasTrustLevel$
- $[P_1\,(A, M_1)\ \wedge\ P_1\,(A, M_2)]\ \wedge\ [P_2\,(M_1, M_2)\ \wedge\ P_3\,\{M_1, (\geq 0.7)\}] \Rightarrow$
  M_1 *gets full access to* M_2'*s private resources*

Definition 2. Limited access to private resources of individuals. M_1 gets limited access to the private resources of M_2.

Definition 2.1. When they both belong to the same community and M_1 has trust level less than 0.7.

- $[P_1\,(A, M_1)\ \wedge\ P_1\,(A, M_2)]\ \wedge\ [P_2\,(M_1, M_2)\ \wedge\ P_3\,\{M_1, (< 0.7)\}] \Rightarrow$
  M_1 *gets full access to* M_2'*s private resources*

Definition 2.2. When they both belong to the different communities but are friend to each other with trust level greater or equal to 0.7.

- $B \in SC\,(Community)$
- $[P_1\,(A, M_1)\ \wedge\ P_1\,(B, M_2)]\ \wedge\ [P_2\,(M_1, M_2)\ \wedge\ P_3\,\{M_1, (\geq 0.7)\}] \Rightarrow$
  M_1 *gets limited access to* M_2'*s private resources*

Definition 3. Access denied to private resources of individuals. M_1 cannot get access to the private resources of M_2 when they both belong to the different communities and also are not friend to each other.

- $P_4 \in P$ where $P_4 = \neg P_2$

- $[P_1(A, M_1) \ \wedge \ P_1(B, M_2)] \ \wedge \ [P_4(M_1, M_2)] \Rightarrow$
  M_1 *denied access to* $M_2's$ *private resources*

Definition 4. Full access to community resources. If M_1 and M_2 belong to the same community, they will get full access to its resources.

- $[P_1(A, M_1) \ \wedge \ P_1(A, M_2)] \Rightarrow$
  $M_1 \& M_2$ *get full access to their community resources*

Definition 5. Limited access to community resources. If M_1 and M_2 belong to the different communities, they will get limited access to each other's community resources.

- $[P_1(A, M_1) \ \wedge \ P_1(B, M_2)] \Rightarrow$
  $M_1 \& M_2$ *get limited access to each other's community resources*

6.2 Policy enforcement through rules and rule engine

The authorisation policy definitions described above are realised using SWRL rules. The rules are formulated using the classes, properties and instances of the community ontology. The Jess rule engine is used to execute these rules and thus to produce authorisation policy decisions. As a first step, the Jess engine converts the relevant OWL knowledge and SWRL rules to Jess knowledge. Then the engine executes the rules and at the end the inferred facts can be exported back to the OWL knowledge base. All these actions are user driven. The rules are formulated as follows according to the policy definitions:

Definition 1: $Member(?personA) \ \wedge \ Member(?personB) \ \wedge$
$hasMember(?Comm, ?personA) \ \wedge \ hasMember(?Comm, ?personB) \ \wedge$
$hasPrivateResource(?personA, ?resA) \ \wedge \ hasFriend(?personA, ?personB) \ \wedge$
$hasTrustLevel(?personB, ?z) \ \wedge \ swrlb:greaterThanOrEqual(?z, 0.7)$
$\longrightarrow hasFullAccess(?personB, ?resA)$

Definition 2.1: $Member(?personA) \ \wedge \ Member(?personB) \ \wedge$
$hasMember(?Comm, ?personA) \ \wedge \ hasMember(?Comm, ?personB) \ \wedge$
$hasPrivateResource(?personA, ?resA) \ \wedge \ hasFriend(?personA, ?personB) \ \wedge$
$hasTrustLevel(?personB, ?z) \ \wedge \ swrlb:lessThan(?z, 0.7)$
$\longrightarrow hasLimitedAccess(?personB, ?resA)$

Definition 2.2: $Member(?personA) \ \wedge \ hasPrivateResource(?personA, ?resA) \ \wedge$
$Member(?personB) \ \wedge \ hasMember(?CommA, ?personA) \ \wedge$
$hasMember(?CommB, ?personB) \ \wedge \ hasFriend(?personA, ?personB) \ \wedge$
$hasTrustLevel(?personB, ?z) \ \wedge \ swrlb:greaterThanOrEqual(?z, 0.7)$
$\longrightarrow hasLimitedAccess(?personB, ?resA)$

Definition 3: $Member(?personA) \ \wedge \ hasPrivateResource(?personA, ?resA) \ \wedge$
$Member(?personB) \wedge hasMember(?CommA, ?personA) \wedge hasMember(?CommB, ?personB) \wedge$
$notFriendOf(?personA, ?personB) \longrightarrow hasNoAccess(?personB, ?resA)$

Definition 4: $Member(?personA) \ \wedge \ hasCommunityResource(?Comm, ?resA) \ \wedge$
$hasMember(?Comm, ?personA) \longrightarrow hasFullAccess(?personA, ?resA)$

Definition 5: $Member(?personA) \ \wedge \ hasCommunityResource(?CommA, ?resA) \ \wedge$
$hasMember(?CommB, ?personA) \longrightarrow hasLimitedAccess(?personA, ?resA)$

The figure 7 shows a rule from the SWRL rule editor. The figure 8 shows the inferred facts after executing the rules using the Jess rule engine. *hasFullAccess(Josef,BillOnt:PrivatePartyVideo1)* literally means, 'Josef has full access to the Private Party Video1 of Bill'. It shows all the facts according to each definition. In practice, query handler generates a SPARQL query for a specific resource provided the requester is authenticated beforehand. In response to this query, only the specific fact is returned in machine processable format.

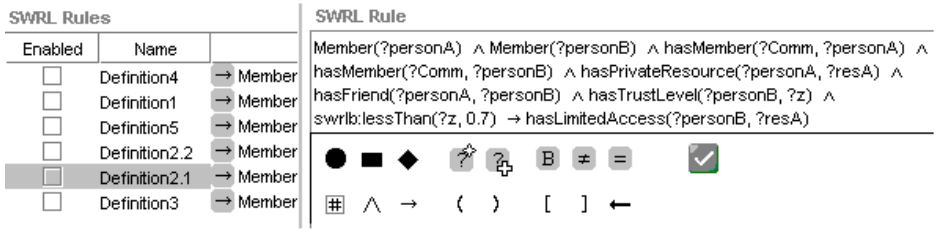

Figure 7 The rules with SWRL rule editor according to the definitions.

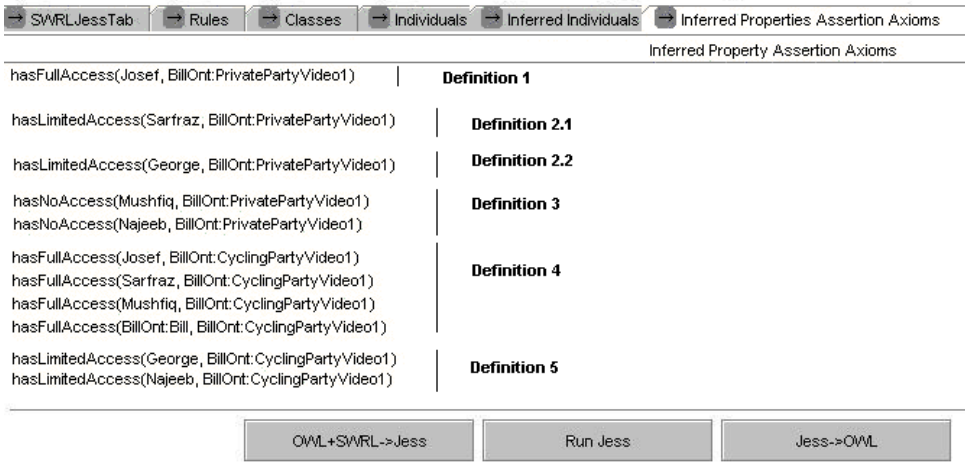

Figure 8 The inferred implicit facts when the rules are executed.

7 Evaluation and related works

The evaluation of the framework focuses on the critical components of it, the Social Semantic Knowledge Base and the Semantic Policy Enforcement Engine. A Semantic Knowledge Base is straight forward to be implemented, thus focus is more on the mediation between ontologies through the mapping. The SPEE is a semantically enhanced Policy Enforcement Engine, and as such the main focus will be on the impact of semantics on the enforcement of policies. Our preliminary conclusion is that the limitations of the underlying description logic and mapping method bring unnecessary complexity to the framework.

Description logic (DL) enhances the expressivity of OWL by facilitating automated reasoning. Reasoning in OWL DL is based on what is known as the open world assumptions (OWA). The OWA means that we cannot assume something

does not exist until it is explicitly stated that it does not exist. Therefore, we have to explicitly define that all instances of the Members are different from each other. Moreover, we also declare that Community, Member and Resource are disjoint concepts so that an individual can not be the instance of more than one of these classes. These limitations remain there even after adding the rules, because SWRL is an extension of OWL DL. Therefore, the W3C suggested that SWRL might gain from being separated from OWL DL. As SWRL does not support negation, we could not derive the property *notFriendOf* through rules and we stated that explicitly in the ontology. It is not possible to remove existing knowledge from the Knowledge Base by using SWRL. Running SWRL is also computationally expensive and decidability is not guaranteed. Despite of these limitations, the use of SWRL along with OWL DL provides more expressivity as the framework using OWL DL and SWRL can infer implicit relations. In this framework, we map the elements of the ontologies manually which is very tedious and inefficient. Besides, such mapping method in real community platform may arise questions from the scalability and usability point of view.

Adding privacy-enhancing technologies (PET) in virtual community networks is subject to research in many communities (Chewar et al., 2003; Walters, 2001). In this part we introduce some of the other initiatives and compare them with our proposed solution. The SIOC (Semantically-Interlinked Online Communities) initiative is particularly aimed at creating ontologies which fully describe the concepts of online community and open for linking to and browsing in these communities (Breslin et al., 2005). Though SIOC-based systems use cross-site queries, privacy features and decentralised architectures of the community are not discussed there. A concept of trust or reputation has been used by Choi (2006) to create and access communities. A distributed trust management was introduced in (Finin, 2002) to provide access to community resources and privacy solutions only by means of trust/reputation management. In this framework, we include the trust through trust metrics while designing the authtorisation policies. Though the metrics are static (hardcoded), the values can be dynamically updated from the distributed trust management approach or can be generated through rules from various context/preference inputs.

In (Li et al., 2006), the authors suggested expressing the access control policies based on OWL and SWRL citing the lack of formal semantics in XACML (eXtensible Access Control Markup Language) which is a popular access control policy language based on XML. KAoS (Bradshaw et al., 1997) and Rei (Kagal) are the two noticeable works in this regard. Policy specification language in Rei is based on OWL Lite which is less expressive compared with OWL DL. In (Javanmardi et al., 2006), a Semantic Based Access Control Model was presented which considered semantic relations among different entities in decision making process. The solutions were limited to the definition of OWL ontology and declaration of SWRL rules without the rule engine support. In this framework, we include the decision support using the rule engine.

8 Conclusion and future work

Online social communities and content sharing sites are one of the most widely used Web application nowadays. Increasing awareness to online confidentiality protection demands privacy enhancement of these applications. The proposed framework is expected to meet these challenges through distributed community mechanism and a fine-grained content access management. The framework allows community members to manage their profiles at own spaces and the disclosure of sharable information or contents to the community. The customised content access constraints are formulated with the combination of multiple relations among the individuals and with the communities. These features are formally represented and later the implicit relationships between the actors are inferred using the promising Semantic Web Technologies.

The paper includes the component descriptions of the functional architecture. As a future work, we are concentrating on the practical realisation of this architecture which has been introduced briefly in chapter 4. It includes the conversion of the community knowledge base into RDF triple data stores and linking it with a SPARQL end point through an interface. Thus it can be accessed by sending SPARQL queries.

Acknowledgements

This work was supported in part by the Norwegian Research Council in the SWACOM project and the ITEA WellCom Project. The authors would like to acknowledge the comments and suggestions of the project partners and the two anonymous reviewers.

References

Abercrombie, N., Hill, S. and Turner, B. S. (2000) 'Social structure'. *The Penguin Dictionary of Sociology*, 4th edition, London: Penguin, pp. 326327, 2000.

Berners-Lee, Tim, James Hendler and Ora Lassila (2001). The Semantic Web. *Scientific American Magazine*, May 17. 2001.

Breslin, J. G. and Decker, S. (2007) The Future of Social Networks on the Internet: The Need for Semantics. *IEEE Internet Computing*, vol. 11, pp. 86-90, November/December 2007.

Breslin, J. G., Harth, A., Bojars, U., and Decker, S. (2005) Towards Semantically-Interlinked Online Communities. *Proceedings of the 2nd European Semantic Web Conference (ESWC'05), LNCS*, vol. 3532, pp. 500-514, Heraklion, Greece, 2005.

Brad Fitzpatrick. (2007) Thoughts on the Social Graph. http://bradfitz.com/social-graph-problem/ [accessed on Feb 22, 2008]

Bradsaw, J. M., Dutfield, S., Benoit, P., and Woolley, J. D. (1997) KAoS: Toward An Industrial-Strength Open Agent Architecture. *Software Agents*, J.M. Bradshaw (ed.), AAAI Press, 375-418.

Bruijn, J. de, Ehrig, M., Feier, C., Martin-Recuerda, F. J., Scharffe, F., and Weiten, M. (2006) Ontology Mediation, Merging, and Aligning. In John Davies, Paul Warren, and Rudi Studer: *Semantic Web Technologies*, John Wiley & Sons, 2006.

Chewar, C. M. D., McCrickard, Scott and Carroll, J. M. (2003) Persistent virtual identity in community networks: Impact to social capital value chains. *Technical Report TR-03-01*, Computer Science, Virginia Tech, 2003.

Chowdhury, Mohammad M. R. and Noll, J. (2007) Integrated Identity Mechanism for Ubiquitous Service Access. *IADIS International Journal on Computer Science and Information System (IJCSIS)*, Vol. 2 No. 2, 2007, pp. 51-64.

Choi, H.-C., Kruk S. R., Grzonkowski, S., Stankiewicz, K., Davis, B., Breslin, J. G. (2006) Trust Models for Community-Aware Identity Management. *Identity, Reference and the Web IRW2006, WWW2006 Workshop*, Scotland, May 23, 2006.

ComScore. (2007) Social Networking Goes Global. http://www.comscore.com/press/release.asp?press=1555 [accessed on Feb. 22, 2008]

Fensel, D. (2004) Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. *Springer-Verlag*, 2nd ed., 2004.

Finin T. and Joshi, A. (2002) Agents, Trust, and Information Access on the Semantic Web. *ACM SIGMOD, Special Issue: Special section on semantic web and data management*, vol. 31, issue 4, December 2002, pp. 30-35.

Hanneman, Robert A. and Mark Riddle. (2005) Introduction to social network methods. Riverside, CA: University of California, Riverside, 2005.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M. (2004) SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission*, May 21 2004.

Javanmardi, S., Amini, M., Jalili, R., Ganjisaffari, Y. (2006) SBAC: Semantic Based Access Control. *The 11th Nordic Workshop on Secure IT-systems*, Linkping, Sweden, October 2006, pp. 157-168.

Kagal, L. Rei Ontology Specifications, Ver 2.0. http://www.cs.umbc.edu/ lkagal1/rei/ [accessed on June 13, 2008]

Klyne, G., Carroll, J. J., and McBride, B. (2004) Resource Description Framework (RDF): Concepts and Abstract Syntax. *W3C Recommendation*, February 10, 2004.

Li, H., Zhang, X., Wu, H., and Qu, Y. (2006) Design and Application of Rule Based Access Control Policies. *International Semantic Web Conference Workshop on Semantic Web and Policy*, 2006, pp. 34-41.

Lu, Jing., Ma, Li., Zhang, Lei., Brunner, Jean-Sebastien., Wang, Chen., Pan, Yue., Yu, Yong. (2007) SOR: A Practical System for Ontology Storage, Reasoning and Search. *Proceedings of the 33rd international conference on Very large data bases*, pp. 1402-1405, Vienna, Austria, 2007.

Motik, B., Sattler, U., and Studer, R. (2004) Query Answering for OWL-DL with Rules. *International Semantic Web Conference 2004, SpringerLink*, pp. 549-563.

McIlraith, S., Son, T. C., & Zeng, H. (2001) Authorization and privacy for semantic web services. *IEEE Int. Systems*, 16(2), 46-53.

Massa, P. and Avesani, Paolo. (2007) Trust-aware Recommender Systems. *Proceeding of ACM Recommender Systems Conference*, Minneapolis, Minnesota, USA, 2007.

Prudhommeaux, E. and Seaborne, A. (2008) SPARQL Query Language for RDF. *W3C Recommendation*, January 15 2008.

Smith, M. K., Welty, C., McGuinness, D. L. (2004) OWL Web Ontology Language Guide. *W3C Recommendation*, February 10 2004.

Walters, G. J. (2001) Privacy and Security: An Ethical Analysis. *Computers and Society*, 2001, pp. 8-23.

Notes

[1] First-order logic (FOL) is a formal deductive system which extends the propositional logic by allowing quantification over individuals of a given domain of discourse.

[2] Logics are decidable if computations/algorithms based on the logics will terminate in a finite time.