

M.Sc. Engg. Thesis

Content-Based Access Control in an Online Community

by

Tasnima Mansoor

Submitted to
Department of Computer Science and Engineering
in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science and Engineering




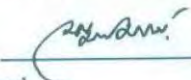
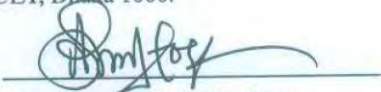
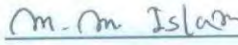

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)

Dhaka-1205

April 18, 2018

The thesis titled “**Content-Based Access Control System in an Online Community**”, submitted by Tasnima Mansoor, Roll No. 1014052054, Session October 2014, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on April 18, 2018.

Board of Examiners

1. 
Dr. Muhammad Masroor Ali
Professor
Department of CSE,
BUET, Dhaka 1000.
Chairman
(Supervisor)
2. 
Head
Department of CSE,
BUET, Dhaka 1000.
(Ex-Officio)
3. 
Dr. Abu Sayed Md. Latiful Hoque
Professor
Department of Computer Science and Engineering,
BUET, Dhaka 1000.
Member
4. 
Dr. Md. Monirul Islam
Professor
Department of Computer Science and Engineering,
BUET, Dhaka 1000.
Member
5. 
Dr. Hasan Sarwar
Professor
Department of Computer Science and Engineering,
United International University, Dhaka.
Member
(External)

Candidate's Declaration

This is to certify that the work presented in this thesis entitled "Content-Based Access Control in an Online Community" is the outcome of the investigation carried out by me under the supervision of Professor Dr. Muhammad Masroor Ali in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka. It is also declared that neither this thesis nor any part thereof has been submitted or is being currently submitted anywhere else for the award of any degree or diploma.

Tasnima Mansoor 08.05.2018

Tasnima Mansoor
Candidate

Contents

Board of Examiners	i
Candidate's Declaration	ii
Acknowledgements	vii
Abstract	viii
1 Introduction	1
1.1 Access Control in Online Social Networks	1
1.2 Research Issues	3
1.3 Organization of the Thesis	3
2 Access Control Methodologies	4
2.1 Relationship-based Access Control Systems	5
2.1.1 ReBAC	5
2.1.2 UURAC	6
2.1.3 URRAC	7
2.2 Rule-Based Access Control Systems	9
2.2.1 RBAC Based on Type, Depth, and Trust Level	9
2.2.2 RBAC: Enhanced Version	12
2.3 Content-Based Access Control System: The MPAC Model	13
2.4 Open Areas in Privacy Mechanisms for Social Network Services	15
2.4.1 Content Type	15
2.4.2 Flexible Relationship Models	15
2.4.3 Usability and Visibility of Relationship Based Access Controls	16
2.4.4 Co-privacy	16
2.4.5 Interoperable Privacy Settings	16
2.5 Addressing of Open Areas in Privacy Mechanisms for Social Network Services	17
3 Semantic Web Technologies	18
3.1 The Semantic Web: A Brief Overview	18

3.1.1	Resource Description Framework (RDF)	20
3.1.2	Ontology and OWL	29
3.2	Semantic Web Logic and Inferences : Rules at a Glance	35
3.2.1	Semantic Web Rule Language (SWRL)	36
3.2.2	SPARQL	37
4	Proposed Framework	39
4.1	Definitions and Brief Overview	39
4.1.1	National Identity (NID):	39
4.1.2	NID Network	41
4.2	Detailed Framework	42
4.2.1	User Profiles	43
4.2.2	Content	43
4.2.3	Privacy Settings	45
4.2.4	Ontology APIs	45
4.2.5	Access Control Rules or Policies	46
5	Methodology Details	47
5.1	Citizens	48
5.2	NID Attributes (Contents)	50
5.3	Interconnection between Citizen and Content	51
6	Content Sharing Controls	52
6.1	Content Sharing in OSN	52
6.1.1	Content Sharing from Own Space	52
6.1.2	Content Sharing in Others' Spaces	53
6.1.3	Sharing Other Users' Contents	54
6.1.4	Tagging Users in Content	54
6.2	Content Sharing in NID Network	55
6.2.1	Required Definitions	55
7	Access Control Policies	58
7.1	Policy Specification	58
7.2	Ontology	60
8	Experiment	66
8.1	Experimental Setup	66
8.2	Cases	68
8.2.1	Common Content	68
8.2.2	Sensitive Content	69
8.2.3	Supervisory Content Settings	70

9	Focused Challenges	72
9.1	Introducing New Attribute: Content Type	72
9.2	Other Focused Challenges	73
9.2.1	Supervisory Settings	73
9.2.2	Sticky Policies and Co-privacy	73
9.2.3	Knowledge Reusing	74
10	Conclusion and Future Work	76
10.1	Compendium of Attainments	76
10.2	Future Work	77

List of Figures

2.1 UURAC Model Components [1].	7
2.2 URRAC Model Components [2].	8
2.3 RBAC System Architecture [3].	11
2.4 Multiparty Access Control Pattern for Content Sharing [4].	13
2.5 Multiparty Access Control Policy Evaluation Process [4].	14
3.1 The RDF Data Model [5].	21
3.2 An RDF statement represented graphically.	21
3.3 An Elaborated RDF Graph.	22
3.4 RDF Data Model Representation with vCard Elements.	24
3.5 Example of a Vocabulary for the Photography Domain [6].	26
3.6 XML datatypes for OWL [7].	33
3.7 The Structure of OWL 2 [8].	35
3.8 SWRLTab Built-In Libraries [9].	37
4.1 Format of NID Card.	41
4.2 Sample Page for NID Network.	41
4.3 NID Service of Election Commission Bangladesh.	44
6.1 Example of Content Sharing in Social Network like Facebook.	53
6.2 Example of Content Sharing in Other User’s Space.	54
6.3 Example of Sharing Other’s Content in Facebook.	54
6.4 Example of Tagging in Social Network.	55
8.1 Class Hierarchy in Protégé.	67
8.2 Graphical Representation of the Class Hierarchy in Protégé. Only up to to sub-classes view has been presented.	67
8.3 Object Properties of the CBAC Ontology in Protégé.	67
8.4 Ontograp Showing Relationship Types and Content Types of CBAC Ontology.	68
8.5 Sample Contents and Individuals for Experimental Cases of CBAC Ontology.	69
8.6 Experimental result validating default sharing for common content.	69
8.7 Experimental result validating default sharing for sensitive content.	70
8.8 Experimental result validating special content settings.	71

Acknowledgements

First of all, I would like to declare that all the appraisals belong to the Almighty.

I would like to express my deep gratitude to my supervisor Professor Dr. Muhammad Masroor Ali for introducing me to the fascinating and prospective field of semantic web and access control using the semantic web technologies. I have learned from him how to carry on a research work, how to write, speak and present well. I thank him for his patience in reviewing my so many inferior drafts, for correcting my proofs and language, suggesting new ways of thinking, leading to the right way, and encouraging me to continue my research work. I again express my indebtedness, sincere gratitude and profound respect to him for his continuous guidance, suggestions and whole hearted supervision throughout the progress of this work.

I convey my heartfelt reverence to my parents and other family members for giving their best support throughout my work to overcome the tedium of repetitive trials to new findings.

Finally, every honor and every victory on earth is due to GOD, descended from Him and must be ascribed to Him. He has endowed me with good health and with the capability to complete this work. I deeply express my sincere gratitude to the endless kindness of the Almighty.

Abstract

This research demonstrates how semantic web can provide a solution for controlling access to contents around a community. The main intention of online communities are enabling a platform for users to share information with each other. But there may be some information to which access must be restricted, such as the personal information of an user in the National ID Network. Traditional OSNs or other online communities provide very few or no controlling access to these information semantically. In this thesis, we have proposed a semantic approach to enhance the access control in any online community depending on the sensitivity of information content. The proposed framework takes into account the information of an individual present against a National ID. An online community is assumed for the sharing of these information based on how sensitive they are for the users possessing the information. We present a detailed framework of the model in a semantic approach and formulate policies for users to control the access of their shared contents. The model is evaluated with some experimental scenarios with synthetic data and the related results have been presented. Moreover, we also mention major challenges such as the complexity of the model and supervisory settings in access control paradigm and show how our model can meet them. Lastly, the proposed model also meets the open area regarding sticky policies and co-privacy related to a shared content.

Chapter 1

Introduction

In this chapter we represent a brief overview of the field we have worked on. At the same time, we also delineate the main focuses of our research. We have presented the outline of our thesis as well in this chapter.

1.1 Access Control in Online Social Networks

In recent years, Online Social Networks (OSN), like Facebook, Twitter, Google+, have become an essential entity to communicate with each other in day-to-day life [10, 11]. The online networks are designed inherently for enabling people to share personal and public information and make connections with friends, coworkers, colleagues, family and even with strangers. But access control to these information has always been an issue of high concern because they can be very sensitive for users. That is why access to shared information must be controlled with some mechanisms. Access control [12] has become a central feature of online communities in order to protect user data.

In our work, we present a model in the context of these information prevalent in online communities. Before going into much detail, here we would like to discuss the type of online community we take as a model. We deal with individual's National Identity or NID information in Bangladesh in this thesis. A National ID Card is a portable document for confirming an individual's national identity. Apart from the physical card, the information is also stored in a database. These information include the person's name, photo, parents' names, address, blood group etc. But besides these basic information, an NID database may also be able to store her

financial, medical or personal information that may be useful for finding more about the specific individual. For our work, we assume that a network is developed for NID holders where they can log into and see their stored information. But they may not want to share these information publicly or share with anyone at all. They may also require to share a subset of these information to selected entities. So, if the NID holders have some controlling options in their hands, just like Online Social Network users, they should feel safe about their sensitive information being shared or not.

We deal with a community of this kind, which we may refer to as “NID Network”. When we say “NID Network”, though it is one type of online community, it will be justifiable to assume that this community’s functionality will not completely be identical to an OSN. Unlike OSN, this community will not have the ‘post sharing’ option due to security reasons. Because if one user is able to share another user’s genuine (we use the word genuine here, because in OSN, users may provide fake information as there is no verification of information; but in NID Network, users must be allowed to publish only verified information) personal content, that piece of content may be the subject to information misuse. So, considering user’s security and privacy issues, sharing of each other’s information is not allowed in our proposed model, though we have allowed some regulatory privilege in applicable cases.

Usually, online communities emphasize on relationships between users for controlling access to shared information and there have been some significant access control mechanisms. Relationship-Based Access Control (ReBAC) [13] is one of the most popular ones which is constructed using user relationships in an online community. Before ReBAC, Rule-Based Access Control (RBAC) [3] took in concern the security of online community basing on the trust level among users. As time went on, there have been many more access control systems [2,4,14–25]). Some of them are based on role, some on user relationships, while some others are based on attributes. An important one is the Multiparty Access Control system [4] that focuses on the security of the shared data, such as, tagged photos, posts etc. among multiple parties. These parties not only attracts genuine, faithful users but also third parties who have adverse interests [25].

The above methods have been discussed in Chapter 2 with their shortcomings for our purpose.

1.2 Research Issues

In our work, we have focused on the type of content instead of relationships. Here, type of content, sensitivity of information to be precise, plays a major role in access control. Based on the type of information, it is decided whether an information is to be made available to all, or to selective users, or not at all. So, the issues handled in this research are precisely as follows:

- Creating working rules and attributes for content sensitivity and sharing. -

Based on the rules, developing policies for content sharing.

- Adopting semantic approach and producing an ontology with required classes and rules in order to enhance access control system.

- Handling co-privacy and sticky policies for dealing with some major challenges of content sharing.

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows: Chapter 2 presents various existing access control mechanisms. Chapter 3 gives a brief idea about the Semantic Web and related technologies for better understanding of the proposed model while it takes the benefits of universal, reusable knowledge. Through Chapter 4 to Chapter 7 we present our proposed model, define the terminologies and notations used in this thesis, discuss the policy specification and rules and present an ontology. Chapter 8 elaborates on some experimental cases which we have performed in order to substantiate the validity of our proposed methodology and reports experimental results on synthetic data sets. Chapter 9 discusses about the challenges focused by the proposed model. Finally, Chapter 10 enumerates the attainments of this thesis and then concludes the thesis suggesting the future extensions possible.

Chapter 2

Access Control Methodologies

Access Control can be defined as an approach to protect user data that individuals share on the web from getting misused by others. It is a set of methods and components that are used for protecting information resources. Although some information is and should be accessible by everyone in the community, users will certainly need to restrict access to other information. Access control implements both the confidentiality and the integrity of a secure system. Eventually, it gives users the ability to determine what information a user can view or modify or both.

While talking about access control methodologies, users in the online communities most frequently take into account their relationship with other users and the contents, type of the contents they share in the network, rules and roles present in the community and so on. Based on these various criteria, we briefly discuss in this chapter a series of access control mechanisms presented through [13]- [26]. Each of these presented mechanisms differ in the way of considering the main element upon which the access control system is based on. We have divided them works into following major categories:

1. Relationship-based Access Control Systems: This is the first and most common one as whenever it comes to access control of contents, users think about the relationships they share with each other in personal.
2. Rule-based Access Control Systems: Apart from relationships, an access control system may also have predefined rules for controlling access over contents while classifying the information as subject and the access requesting user as an object.

3. Content-based Access Control Systems: Despite having very few works on it, this one is a major category in access control as the protection of content depends on how sensitive it is to the user(s) involved with it.

Let us now discuss briefly the above mentioned categories highlighting some important works in these fields.

2.1 Relationship-based Access Control Systems

2.1.1 ReBAC

One of the most popular and general-purpose access control model is the Relationship-based Access Control (ReBAC) model introduced in [13]. This model highlights on poly-relational OSNs with irregular relationships, while also features a context-dependent authorization procedure limiting the scope of relationships to their applicable contexts. A group of OSNs form the state of a ReBAC system. In this model, state transition involves variation of the social networks, such as addition or deletion of relationship edges. Three terms are included in state transition: accessor, resource and owner. When someone attempts an access at a given state, the ReBAC policy regulating the accessibility of the requested resource will be evaluated against a certain social network induced by the current state. A twofold relationship is defined by the ReBAC policy between the owners and accessors of a resource in the context of a given OSN.

This work demonstrates that there are known relational policies. Rather than relating policy families to enforcement mechanisms, this work establishes upper bound and lower bound for the family of policies definable in a policy language. This language is an extended ReBAC policy language E .

In the extended policy language, a new connective has been introduced to acquire disjoint intermediaries and vertex identification. The ReBAC model shows that:

ReBAC policies definable in E are owner-checkable policies, a superset of relational policies.

This extended policy language is able to define every finite relational policy and so is complete with that respective policy family.

The policy language is adequate for expressing all useful ReBAC policies.

2.1.2 UURAC

Because of the significance of user-to-user relationships in specifying and enforcing access control in OSNs, the UURAC (User-to-User Relationship-based Access Control) model has been proposed in [1] for specifying policies regarding grant of access to resources. This model allows individual users and OSN providers to specify which access can be granted in terms of existing relationships. For specifying such policies, regular expression notation is used in the UURAC model. The model supports policy individualization as well as user and resource as a target. It also holds the distinction of user policies for outgoing and incoming actions, and relationship-based access control. The followings are incorporated in UURAC:

Better generalization of path patterns in policy specification.

Inverse relationships between users.

Effective path checking algorithm for evaluation of access control policy [1].

The user relationship path in access control policies of the UURAC model is represented by regular expressions. A DFS-based algorithm has been used for path checking. Each specification describes a pattern of required relationship types between the accessing user and the target/controlling user. Here, target is the recipient of an action, i.e., an abstract function initiated by an accessing user (u_a) against a resource. A target can be either an user (u_t) or a resource (r_t). According to the UURAC model, a target user has U2U (User-to-User) information along with her own policies, both of which are used for authorization decisions. On the other hand, target resource has U2R (User-to-Resource) relationship (i.e., ownership) with the controlling users (u_c) of the resource. For accessing a target resource, an accessing user must possess the required U2R relationship with the controlling user. When the accessing user requests a certain type of action against a target, it is called access request according to the UU-RAC model. For access authorization, the model introduces policies that define rules according to which authorizations are regulated. So, the five major components of the UURAC models are as shown in Figure 2.1.

The model separates user policies for incoming and outgoing actions. For example, when a user wants to block a particular friend's activity-notifications, the regarding policy is considered as incoming action policy. On the other hand, policies for controlling own or other users' activities are captured as outgoing action policies. By specifying these policies, the UURAC model

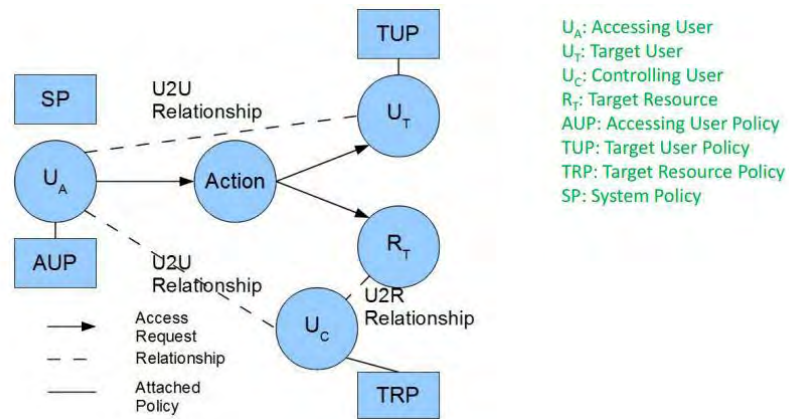


Figure 2.1: UURAC Model Components [1].

can give a flavor of parental control to the users of the OSNs. The policies are represented by regular expressions. Three kinds of wildcard notations are used to represent different relationship type occurrences: asterisk (*) for 0 or more, plus (+) for 1 or more and question mark(?) for 0 or 1.

An access control policy in the proposed model also requires graph rule. Each path spec consists of a pair (path, hopcount), where path is a sequence of characters, denoting the pattern of relationship path between two users that must be satisfied, while hopcount limits the maximum number of edges on the path.

While the UURAC model proves the correctness of the algorithm for relationship path between users for a given access request, the model needs more flexible grip over parental policies and related user's control (e.g., tagged user). The future direction of the work includes incorporation of some predicate expressions for attribute-based control and filtering users and relationships while capturing unconventional relationships in OSNs at the same time.

2.1.3 URRAC

The limitations of the UURAC model led path to User-Resource relationship based access control [2]. This work focuses beyond User-to-User relationship. While UURAC model generalizes the path pattern in policy specification, URRAC makes it more expressive through different relationship types and hopcount skipping. Another advantage is the system-level conflict resolution policy [2]. Followings are the characteristics of URRAC in OSNs:

Policy Individualization:

- Users are allowed to define their own privacy and activity preferences.

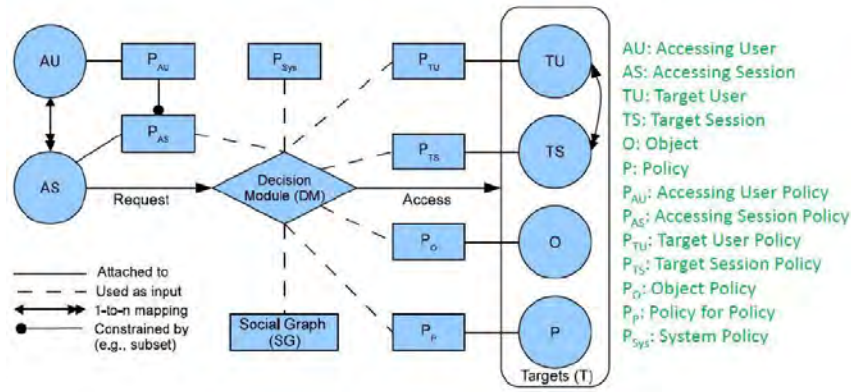


Figure 2.2: URRAC Model Components [2].

- Related users have the privilege to configure policies as well.
- Collectively used by the system for control decision.

Policy Administration:

- Policy and Relationship Management.
- Users specify policies for other users and resources.

User-session Distinction:

- Multiple sessions for a single user is allowed with different sets of privileges.
- Quite useful in mobile and location-based applications.

The components of the URRAC model can be viewed in Figure 2.2.

In this model, OSN supported actions are defined as a set, $ACT = \{act_1, act_2, \dots, act_n\}$. Access request on a target T is defined as a tuple, $\langle s, act, T \rangle$, where s tries to perform act on T and $T \subseteq (2^{TU} \cup R - \emptyset)$ is a non-empty set of users and resources which may contain multiple targets. The reverse of an action is defined as $action^{-1}$ which is the passive form since it applies to the recipient of action. Authorization policies of the URRAC model are shown in Table 2.1.

Here, System Policy (SP) does not differentiate the active and passive forms and o.type is needed to refine the scope of the resource in SP for resource. The main feature of the UR-RAC model is hopcount skipping that enhances the expressiveness of the model. In hopcount skipping, distance created by resources between users are omitted because U2R and R2R relationships may form a long sequence. Policies in this model are evaluated as a combined result

Accessing User Policy	$\langle act, graphrule \rangle$
Accessing Session Policy	$\langle act, graphrule \rangle$
Target User Policy	$\langle act^{-1}, graphrule \rangle$
Target Session Policy	$\langle act^{-1}, graphrule \rangle$
Object Policy	$\langle act^{-1}, graphrule \rangle$
Policy for Policy	$\langle act^{-1}, graphrule \rangle$
System Policy for User	$\langle act, graphrule \rangle$
System Policy for Resource	$\langle act, o.type, graphrule \rangle$ where <i>o.type</i> is optional

Table 2.1: URRAC Authorization Policy [2].

based on conjunctive or disjunctive connectives between path specs. For this, a collective result for multiple policies is formed in each policy set. Then the final result is composed from the result of each policy set. For avoiding policy conflict, system-level conflict-resolution policies are provided based on relationship precedence.

2.2 Rule-Based Access Control Systems

2.2.1 RBAC Based on Type, Depth, and Trust Level

In [3], a rule-based approach is presented where a subject requesting access to an object must demonstrate that it has the rights of performing that action. In this model, policies are expressed as constraints on the type, depth, and trust level of existing relationships. The main features are the use of certificates for granting relationships' authenticity, and the client-side enforcement of access control according to the rule-based approach.

In this approach, the client-side and decentralized access control are proposed to be customized in the context of web based social networks. Here, a user who requests a resource is defined as requestor while the user who possesses the resource is known as the resource owner. When a requestor requests a resource to a resource owner, the former receives from the latter a set of access rules which manage the release of the requested resource. These rules basically state which type of relationship should exist between the resource owner and the requestor, and the maximum depth and minimum trust level allowed.

The requestor needs to yield a proof to the resource owner establishing that there exists the required relationship between them, and that this relationship possesses the obligatory depth and trust level. With this proof, the resource owner is able to demonstrate locally if (s)he

wants to release the resource or not. It is to be mentioned that in this model, access rules are defined using N3 and evaluated by the Cwm reasoner. Cwm (pronounced coom) is a general -purpose data processor for the semantic web, proposed by Tim Berners-Lee [27]. Being a forward chaining reasoner, Cwm can be used for querying, checking, transforming and filtering information. It uses RDF in an extended form in order to include rules and uses RDF/XML or N3 logic [28]. The Cwm reasoner is used in [3] for verifying the validity of corresponding proof of the rules in the privacy mechanism. Here, the following issue needs to be solved which questions the way of a resource owner's being ensured that assertions created by the requestor are appropriate and trustworthy.

One of the proposed solutions is the notion of certification according to which, whenever two users establishes a new relation, both of them shall create and sign a certificate yielding that there exists a direct relationship between these users with an assured trust. But another issue arises with this solution. That is, the trust level between two nodes is figured out considering all the possible paths establishing a relationship between them. Which states that justifying the allegation on the relationship's trust requires a complicated strategy.

To overcome this problem, a semi-decentralized architecture is proposed. According to this architecture, a given trusted node is responsible for managing certificates, and of computing the trust levels of relationships. This node is referred to as central node CN. Upon receiving the access rules from a resource owner that regulates the release of his/ her resource, the requestor requests to CN the certificate chains that prove the existence of a given relationship along with its trust level. If there exists any certificate chain, it is then generated by CN after consulting the certificate archive. But before being returned to the requestor, the corresponding trust level is signed by CN. The received certificate chain is then used by the requestor for generating the assertion for the reasoner.

The main advantage of such semi-decentralized solution is that it fits well with the architecture of current social networks. In the semi-decentralized approach, the central node is responsible for the tasks usually done by the Social Network Management System (SNMS), while the other nodes are in charge of enforcing access control for the resources under their possession. It is also to be mentioned that CN is not in charge of storing users' personal data or anything else other than the relationship certificates only. In this way, users' data confidentiality is ensured with respect to CN.

According to the system architecture, there exists two main services: the central node CN,

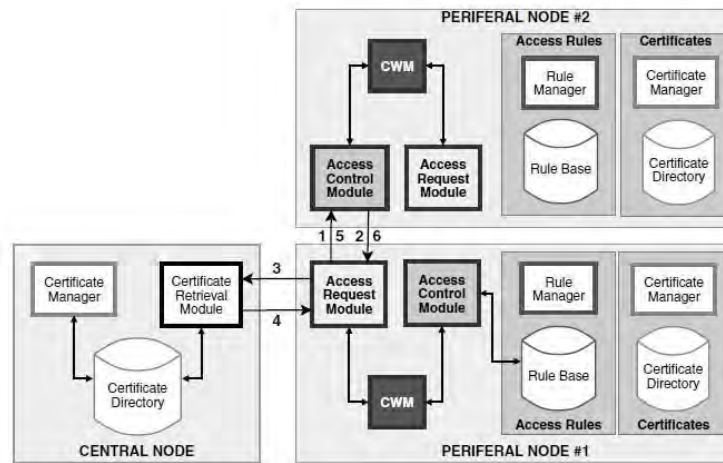


Figure 2.3: RBAC System Architecture [3].

corresponding to the SNMS, and a set of peripheral nodes, corresponding to the nodes in the network. Users in this system are identified by OpenID addresses as it exploits the OpenID framework. Every information accessible via Web is considered as a resource and is identified by a URI.

The central node consists of the certificate repository that stores all the user-generated certificates and of two fundamental modules: the certificate manager and the certificate retrieval module. The former one is liable for receiving the user-generated certificates in the network. It also verifies their validity at the same time performing some modifying operations. On the other hand, the latter one, i.e., the certificate retrieval module replies to certificate chain requests which exist between two network nodes. This module figures out the trust level of a relationship as well by judging the stored information into the respective certificate chain. The main components of the system is shown in Figure 2.3.

Peripheral nodes are services that must be run by a sever machine. These nodes consist of four major components: the certificate manager, the rule manager, and the access control and access request modules. As can be seen, just as the central node, peripheral nodes also have a certificate manager. This certificate manager stores and manages the user-generated certificates while the rule manager is in charge of storing and managing all the access rules specified by a resource owner for his/her resources.

The access control manager (ACM) is responsible for replying to the access requests which are submitted by users in the network. On the other hand, the access request module (ARM) performs the actions as follow: a) submits access requests, b) retrieves the certificate chains

from the central node and produces the relationship assertions to be used in a proof, and c) computes the proof itself. Both ACM and ARM accomplishes their own tasks using the N3 translator and the Cwm reasoner.

It is to be noted that ARM communicates only with the central node and with the ACMs of other peripheral nodes. That is why it is considered to be a quite independent component of a peripheral node.

This model, having features like certificate-used authenticity and client-side enforcement of access control using a rule-based approach, intends to introduce some more attributes in future works such as support for topical trust and the usage of access rules also for certificate protection.

2.2.2 RBAC: Enhanced Version

This model presented in [23], also follows a semi-decentralized architecture and carries out client-side access control enforcement. But unlike the one mentioned in Section 2.2.1, it provides the detail of access control enforcement besides focusing on the access control model.

To cope with the issue of verifying trustworthiness at client-side, this model proposes a solution based on the notion of relationship certificates. Introducing a certificate server enhances the efficiency of the overall framework. The relationship certificate is used for expressing a relationship in the proposed model. The certificates, being generated and signed, are uploaded to a directory, CCD, which is the certificate directory of the certificate server. This process is done after having checked the validity of their signatures. Copies of such certificates are also held by their generating nodes into the local certificate directories. There is a Certificate Revocation List in the certificate server for managing certificate revocation.

The certificate path discovery is performed through discovering the shortest certificate paths that refer to the set of access conditions that are received by the requestor node. Exploration of network graph helps in achieving this. For example, an algorithm exploiting BFS architecture is used for exploring the social network graph and discovering the certificate paths satisfying a given access condition. But this task may have high computational cost, depending on the degree and the order of the graph itself.

Finally, for dealing with security attacks such as unauthorized access or denial of service, standard strategies are intended to be adopted by this model that are used by online systems in

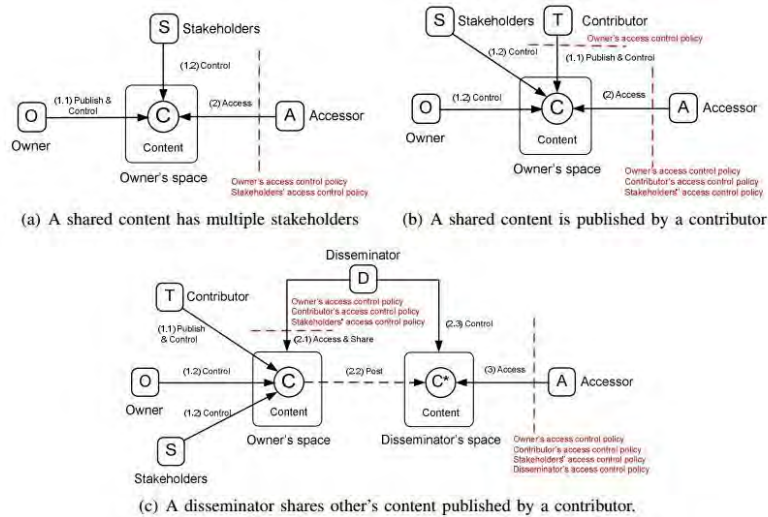


Figure 2.4: Multiparty Access Control Pattern for Content Sharing [4].

order to reduce the risk of attacks.

2.3 Content-Based Access Control System: The MPAC Model

Among the very few content-based access control systems, the Multiparty Access Control model described in [4] is one of the most significant works. This model intends to capture the structure of multiparty authorization requirements. It also formulates a multiparty policy specification scheme and a policy enforcement mechanism for controlling access control in online communities.

The MPAC Model allows multiple controllers associated with shared data to specify access control policies. According to this model, the contributor, stakeholder and disseminator of data, need to regulate the access of the shared data along with the data owner, as can be seen in Figure 2.4.

This model also introduces sensitivity levels to deploy effective privacy conflict resolution for multiparty access control. These sensitivity levels are assigned to the shared data items by the controllers.

For evaluating an access request over multiparty access control policies, this model introduces two steps. In the first step, the access request against the policy is checked which is specified by each controller. At the end of this step, a decision for the controller is yielded. The accessor element in a policy decides whether the policy is applicable to a request or not. If the request sender belongs to the user set that is derived from the accessor of a policy, the

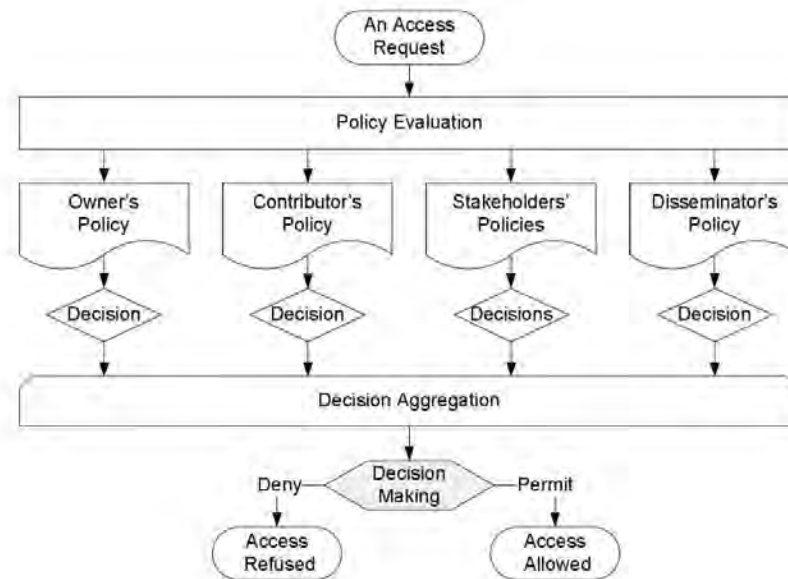


Figure 2.5: Multiparty Access Control Policy Evaluation Process [4].

policy is applicable and the evaluation process returns a response with the decision (either permit or deny) indicated by the effect element in the policy. The response returns deny decision if the policy is not applicable to the request. The next step of the MPAC policy evaluation accumulates the decisions from all controllers responding to the access request for yielding a final decision for the access request. Figure 2.5 shows the policy evaluation process for multiparty access control.

In order to provide a better privacy protection, the MPAC model deploys a strong privacy conflict resolution strategy by introducing a simple and flexible voting scheme that contains Decision Voting and Sensitivity Voting. The first one follows a weighed decision voting scheme while the latter one works on the base of sensitivity score(SC) ranging from 0.00 to 1.00.

Apart from the above described ones, there are many more access control systems which are also based on attributes, or roles, or both. A ciphertext-policy attribute-based model is proposed in [15], where encryption is used for enforcing access control policies with efficient attribute and user revocation capability. The fine-grained access control is achieved in this model by dual encryption mechanism getting benefited from the attribute-based encryption and selective group key distribution in each attribute group. Another attribute-based model is described in [19], which combines and overcomes the limitations of three different access control models including RBAC.

An ontology-based access control model, OSNAC, is described in [18], which is designed upon the Semantic Web technologies (described in Chapter 3) taking into account the complex-

ity of some semantic relations among data objects, different users, and between data objects and users. The key idea in OSNAC is expressing the policies on the relations among concepts in the social network ontology. In this model, policies are provided for the system to define an authority model for deciding which users' policies are effective on what protected resources.

2.4 Open Areas in Privacy Mechanisms for Social Network

Services

Although the access control systems discussed in the previous sections fulfill the requirements for controlling access over the shared information in the OSNs, there are many probable lines for research. According to [29], some of the major open areas in the existing privacy mechanisms for the online communities are:

2.4.1 Content Type

Current access control models do not consider content type while users in the online communities tend to take into account the type of the content they share. Introducing a new attribute, content type, may certainly improve the expressiveness and flexibility of the access control model. On the other hand, more attributes can be responsible for increased complexity of the privacy mechanism and access control. So it is a matter of consideration whether to include content type as a new attribute of the access control or not.

2.4.2 Flexible Relationship Models

According to [26], the disclosure of private information represents an important part of these relationships. The way users share information with each other changes the perception of both parts about the way they should interact in the future. A self-disclosure decision-making mechanism can come handy in this regard. The work in [30] proposes such a privacy mechanism for multiagent systems that takes into account the psychological findings regarding how humans disclose personal information in the building of their relationships. The implementation of such mechanisms to a relationship model would definitely increase the accuracy of that model over time [29].

2.4.3 Usability and Visibility of Relationship Based Access Controls

Relationship based access control models, while using technologies like semantic web or regular expressions, should be able to hide the complexity of such technologies and facilitate their use. Again, inclusion of new attributes like content type are also supposed to have alteration in the existing privacy control mechanism. As a result, more usable and flexible privacy controls need to be created for further access control in near future.

Similarly, complex models that allow the specification of a privacy policy hierarchy, user-to-resource relationship, or social paths will require visualization tools with intelligent architectures. The visualization tools should be able to explain to the users in an understandable way how their information is shared according to a specific type of relationship. We can mention Google+ and the friend circle application as a good example of a usable and user-friendly design of visualization tool.

2.4.4 Co-privacy

Co-privacy can be liable for several privacy conflicts among the users involved in the sharing of information. A shared item may certainly cause tension among the involved parties. Online communities should offer integrated solutions for situations like these. The consequence of such tense situations may lead to users adopting strategies like un-friending the user who uploaded the controversial item or deleting their profile from the OSN. As a matter of fact, it is necessary to add shared item or co-privacy management to the existing access control models for online communities including OSNs.

2.4.5 Interoperable Privacy Settings

With the increasing number of OSNs, it is very natural for a user having profiles on more than one OSN. Setting up privacy settings for each different OSN profile of the same individual is quite difficult and of course, makes users struggle with it. The task would be less tiresome for users if the access control models for various online communities use a universal and well-defined privacy mechanism ontology, one of the most important semantic web technologies allowing reuse of knowledge. Also, personal privacy agents, if developed, can add more flexibility to the interoperability of privacy settings.

The accomplishment of the above mentioned challenges and more at [29], such as sticky policies and relationship or tie strength, will of course have a great impact on the online communities making them more useful, entertaining and privacy-safe. While users at the online communities will feel safer and hence more encouraged to use services like these, the utility of such online communities will increase since users will publish and share much more information in their profiles.

2.5 Addressing of Open Areas in Privacy Mechanisms for Social Network Services

The open areas described in Section 2.4 have been addressed in this thesis as detailed below:

1. The issue of content type (Section 2.4.1) is the major one that is dealt in this thesis. This issue is addressed throughout Chapter 4 to Chapter 7.
2. Co-privacy and sticky policies (Section 2.4.4) are dealt in Section 9.2.2.
3. Interoperable privacy settings (Section 2.4.5) is dealt in Section 7.2.

Before going into detail of the mentioned addressing, let us now proceed to know the basics of Semantic Web in the following chapter as the proposed model in this thesis chooses to adopt semantic approach.

Chapter 3

Semantic Web Technologies

Semantic Web, introduced first by Tim Berners-Lee [6], is an extension of the traditional Web and is also known as the Web of Meaning. It provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [31]. In our work, we have formulated our model on semantic web technology using ontology. We chose to work with the semantic web technology because of its comprehensiveness and conformity. Here we have presented only those parts of semantic web technology which are pertinent to our work. A complete depiction is out of the scope of this thesis and interested readers would like to consult standard textbooks like [6, 32, 33].

3.1 The Semantic Web: A Brief Overview

Semantic Web was formally introduced to the world as below:

“The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation” [34].

The goal of Semantic Web is not to replace the current Web. Its general vision can be summarized in the following single phrase: ‘to make the web more accessible to computers’ [32]. Because computers play limited role on the traditional web, rational works like selecting, combining, aggregating etc. are to be done by human readers. This is why the idea of Semantic Web emerged in order to make the traditional web richer with information and machine understandable data.

The design principles of Semantic Web requires the following technologies and standards:

1. Resource Description Framework (RDF),
2. Domain specific ontologies/ vocabularies,
3. Web Ontology Language (OWL).

Except the above mentioned ones, there are also some more design principles for Semantic Web, but these are all we need now for getting a complete understanding of this research. The terms mentioned above are described in Section 3.1.1 and 3.1.2.

Before going into the detail of the design principles, let us gather some more knowledge about the concept of Semantic Web. There are two concepts that are closely related to the Semantic Web. They are:

Linked Data, and

Web of Data.

Linked Data is a W3C-backed movement that focuses on connecting data sets across the Web, and it can be viewed as a subset of the Semantic Web concept, which is all about adding meanings to the Web [6]. Tim Berners-Lee originally proposed the idea of Linked data. His 2006 Linked Data Principles [35] is considered to be the official and formal introduction of the concept itself. The basic idea of Linked Data concept refers to publishing structured data online and adding links among these datasets. The datasets are published by using the Semantic Web technologies and standards, while they are linked together also using these technologies and standards of Semantic Web. Which means that there is, of course, a strong relationship between Linked Data and the Semantic Web, where Semantic Web is the goal and Linked Data shows the path to reach the goal. This makes Linked Data to be considered as a set of best practices for publishing and connecting structured data on the Web [6].

Let us focus on the next important concept: Web of Data. It can often be interchanged with the term Semantic Web. At this point, Web of Data can be expressed like this: if Linked Data is comprehended using the standards and technologies of Semantic Web, we would get a Web of Data in result. So, we can consider Semantic Web as a set of standard technologies to realize a Web of Data [6].

Having the basic concepts of Semantic Web cleared, let us now discuss its design principles as follows:

3.1.1 Resource Description Framework (RDF)

The Resource Description Framework (RDF) can be viewed as the building block for the Semantic Web. It was originally created in early 1999 by W3C as a standard for encoding metadata with a history of metadata from 1995. RDF has features that allows data merging through underlying schemas even if they differ with each other [36]. The transformation of schemas are reinforced by RDF without requiring to change all the data consumers. RDF can be viewed as a collaborative design structure with a data model that directly helps to to promote interoperability between applications exchanging machine-understandable information on the Web [5, 6].

Any exchange language has the following three components: a data model, a syntax, and a semantics [32]. RDF, being an exchange language, is not exceptional. Let us discuss briefly these components of RDF.

3.1.1.1 RDF Data Model

The organization or structure of the data is expressed through a data model. For the Semantic Web, a richer data model is required which can be used by multiple applications. This model will not just describe documents for people but will also be describing application-specific information. Moreover, the data model needs to be domain independent so that applications ranging from real estate to social networks can take advantage of this model.

RDF provides just such a flexible domain independent data model for describing resources. A resource in RDF is defined as any object that can be uniquely identified by an Uniform Resource Identifier (URI) [37]. Resources have attributes (or characteristics) known as properties. The properties associated with resources are identified by property-types having corresponding values. In RDF, values may be atomic- such as, text strings, numbers, dates, or other resources, which in turn may have their own properties. When we say description in RDF, we mean a set of these properties referring to the same resource. At the core of RDF is a syntax-independent model that represents resources and their corresponding descriptions [5].

Figure 3.1 illustrates a generic RDF description.

The application and use of the data model above in Figure 3.1 can be well understood by concrete example. Take an example statement “BUET is located in Dhaka.” Here, the statement has two resources: BUET and Dhaka, a property-type of location having the value Dhaka.

In Figure 3.2, labeled nodes are connected by labeled arcs. The arcs are directed from

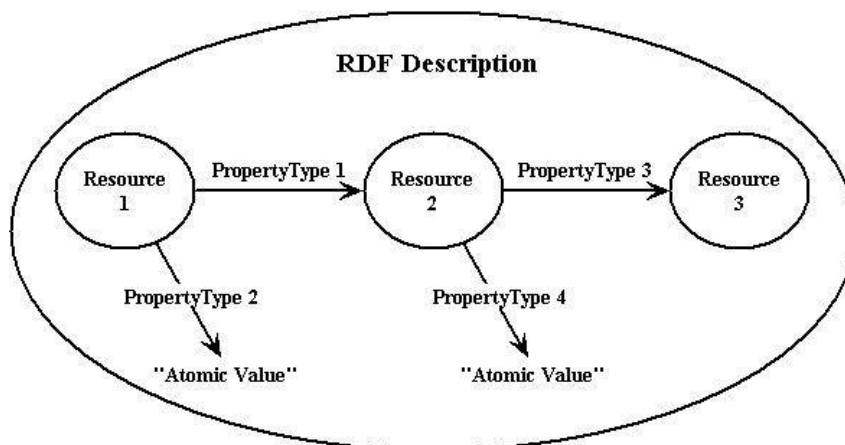


Figure 3.1: The RDF Data Model [5].

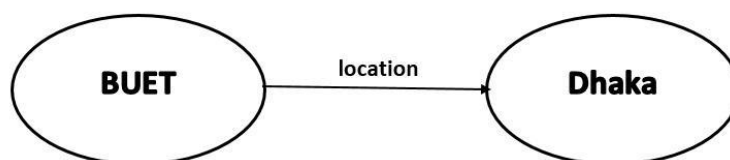


Figure 3.2: An RDF statement represented graphically.

the subject of the statement to the object of the statement, with the label on the arc to the statement's property. The labels on the nodes are the identifiers of the subject and object. The object of a statement can be the subject of another statement. For example, let us now work with a bit complex statement that says, "John Smith is the author of Book1 and his email ID is smith@home.com". RDF breaks this statement down as two statements:

1. "John Smith is the author of Book 1".
2. "John Smith's email ID is smith@home.com".

The graphical representation of this statement will look like Figure 3.3.

We see in Figure 3.3 that instead of the author's name, Book 1 resource points to another resource 'Author_001'. Before descriptive properties can be expressed about the person John Smith, there needs to be a unique identifiable resource representing him for unambiguous association of properties [5]. That is why 'Author_001' is used instead of string value 'John Smith' where 'Author_001' denotes a uniquely identified resource for the author with the property-types of name and email.

While these statements are significantly more complex, the same data model is applicable.

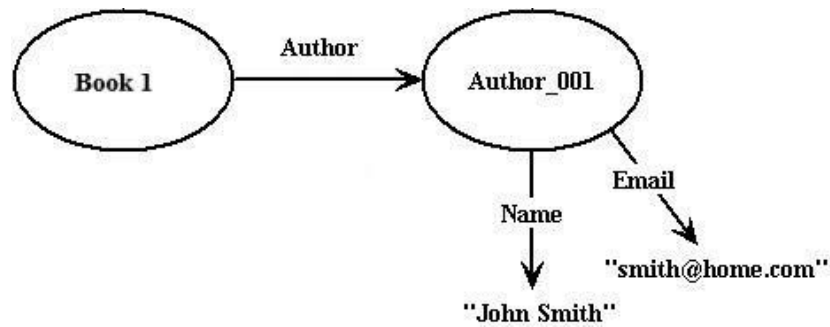


Figure 3.3: An Elaborated RDF Graph.

A more detailed discussion of these issues is outside the scope of this research but more information is available at [36].

3.1.1.2 RDF Syntax

Except from the graphical syntax presented in Section 3.1.1.1, RDF uses more standard, machine interpretable syntax. XML is the elemental syntax for the Semantic Web. A formal structure is imposed on XML by RDF for supporting the consistent representation of semantics.

For preventing ambiguous use of property-types, RDF uniquely identifies property-types through the XML namespace mechanism [38]. For example, let us take in consideration the ‘author’ property-type. The Dublin-Core Initiative [39] defines this property-type as the ‘person or organization responsible for the creation of the intellectual content of the resource’. As a result the ‘author’ property-type is specified by the Dublin Core CREATOR element [40].

Implying the discussion above, the syntax for expressing the statement “John Smith is the author of Book 1” using XML namespaces to identify the use of the Dublin Core Schema can be viewed as follows:

```

1 <?xml:namespace ns = "http://www.w3.org/RDF/RDF/" prefix = "RDF" ?>
2 <?xml:namespace ns = "http://purl.oclc.org/DC/" prefix = "DC" ?>
3
4 <RDF:RDF>
5 <RDF:Description RDF:HREF = "http://uri-of-Book-1"> 6
  <DC:Creator>John Smith</DC:Creator>
7 </RDF:Description>
  
```

8 </RDF:RDF>

From the code listed above, we can say the following points:

1. The URI associated with the namespace declaration references the corresponding schemas.
2. The element <RDF:RDF> is a simple wrapper for marking the boundaries in an XML document where the content intends to be able to get mapped into an RDF data model instance.
3. The element <RDF:Description> is correspondingly used to for denoting a resource with the corresponding URI `http://uri-of-Book-1`.
4. Finally, the <DC:Creator> element represents a property-type <DC:Creator> and a value of 'John Smith' in the context of the <RDF:Description>. The syntactic representation is designed to reflect the corresponding data model.

It is to be mentioned that the Dublin Core Schema is declared in order to utilize the vocabulary required for expressing the RDF data model. But Dublin Core doesn't have definitions for the property-types 'name', 'email' and other additional property-types except for the ones mentioned in [40]. For elements like these, an additional resource description standard may be utilized for feasibility. The vCard representation [41] comes handy for these property-types. Using the vCard representation, the complete statement stating "John Smith is the author of Book1 and his email ID is smith@home.com" has the following syntactic view along with the graphical data model presented in Figure 3.4.

```
1 <?xml:namespace ns = "http://www.w3.org/RDF/RDF/" prefix = "RDF" ?>
2<?xml:namespace ns = "http://purl.oclc.org/DC/" prefix = "DC" ?>
3<?xml:namespace ns = "https://www.w3.org/Submission/vcard-rdf/"
4prefix = "v" ?>
5
6 <RDF:RDF>
7 <RDF:Description RDF:HREF = "http://uri-of-Book-1">
8<DC:Creator RDF:HREF = "#Creator_001"/>
9 </RDF:Description>
```



```

10
11 <RDF:Description ID="Creator_001">
12   <v:Name>John Smith</CARD:Name>
13   <v:Email>smith@home.net</CARD:Email>
14 </RDF:Description>
15 </RDF:RDF>

```

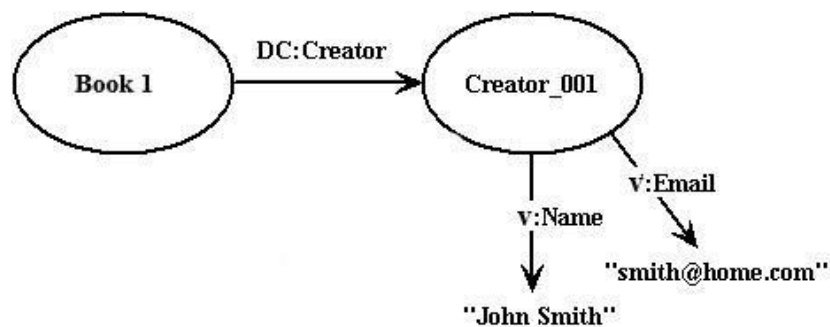


Figure 3.4: RDF Data Model Representation with vCard Elements.

The RDF syntax isn't bound to the Dublin-Core syntax only. It has more syntaxes like Turtle, RDFa and many more. For in-depth knowledge in the RDF syntax, interested readers can be recommended to have a glance through [42–44].

3.1.1.3 The RDF Schema

Let us start with a definition of the RDF Schema (RDFS) in plain English:

“RDFS is a language one can use to create a vocabulary (often the created vocabulary is domain-specific), so when distributed RDF documents are created in this domain, terms from this vocabulary can be used. Therefore, everything we say, we have a reason to say it” [6].

So what we get from the definition is that the RDF Schema is used for declaring vocabularies, i.e., the collections of semantics property-types that are defined by a specific community. Officially, RDFS is a recommendation from W3C and it is an extensible knowledge representation language that can be utilized for creating a vocabulary for explaining classes, subclasses and properties of RDF resources.

The most important fact about RDFS is that it provides a common language which can be used by everyone for defining classes and properties for a specific application domain [6].

Being similar to RDF and Dublin Core terms, the RDFS terms are also identified by predefined URIs sharing the prefix string: `http://www.w3.org/2000/01/rdf-schema#`. The namespace prefix `rdfs:` is associated with the mentioned prefix URI string by tradition.

A simple example can make things more clear. But before that, let us first know the basic groups of the RDFS terms that they can be divided into [6]:

Classes:

When in RDF, we talk about specific objects, in RDFS we talk about the type of these objects. Such as, we stated about Dhaka city in our first example of RDF. But now we want to talk about cities, countries, and so on. Here comes the idea of classes. A class can be viewed as a set of elements of same type. In RDFS, we use the following terms for defining classes:

- `rdfs:Resource`
- `rdfs:Class`
- `rdfs:Literal`
- `rdfs:Datatype`

Properties:

Individual objects that belong to a class are referred to as instances of that particular class. RDFS helps to establish relationship between instances and classes using the following terms:

- `rdfs:range`
- `rdfs:domain`
- `rdfs:subClassOf`
- `rdfs:subPropertyOf`
- `rdfs:label`
- `rdfs:comment`

Utilities:

The terms included in this group are used for miscellaneous purposes. For now, it is enough to know that this group includes `rdfs:seeAlso` and `rdfs:isDefinedBy`.

Let us consider a small domain of ‘photography’ that seems like Figure 3.5.

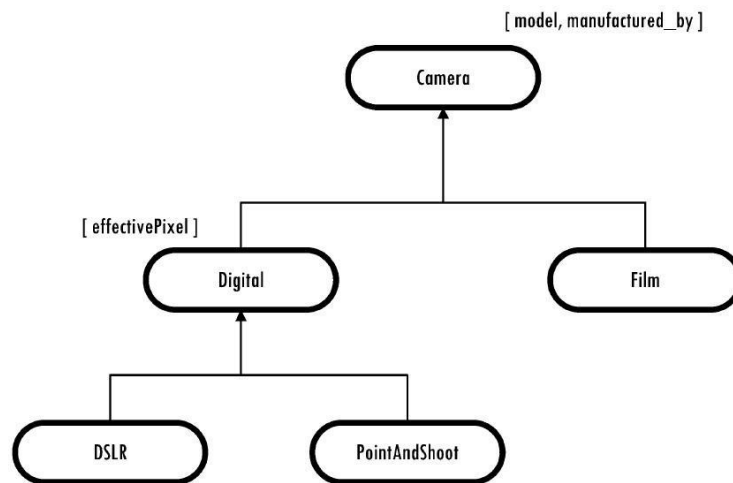


Figure 3.5: Example of a Vocabulary for the Photography Domain [6].

From the figure above, we get to know the following issues:

1. Oval boxes are used for representing specific resource types.
2. Arrows are used to connect these oval boxes. A connecting arrow between two oval boxes means that the first oval box is a subtype of the second oval box.
3. Third brackets ([]) include the properties which can be used for describing a given re-source type and are placed beside the particular oval box.

So now the simple vocabulary presented in Figure 3.5 gives the following facts as output for us:

There is a resource called **Camera**, which has two subresources **Digital** and **Film**.

Again, **Digital** has two more resources under it as subdomains, namely, **DSLR** and **PointAndShoot**.

We can describe the resource **Camera** by properties called **model** and **manufactured_by**. Similarly, resource **Digital** can be described by a property called **effectivePixel**.

At this point, let us keep in mind that DSLR cameras are likely to be used by professional photographers while Point and Shoot, or compact cameras tend to be used by non-professionals and are cheaper than the DSLR ones. Now, using the vocabulary above, if we want to describe

the Canon PowerShot ELPH 160 as a Point and Shoot Camera having 20.0 Megapixels, we will have a list like the one below:

```
1 <?xml version="1.0"?>
2   <rdf:RDF
3     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5     xmlns:myCamera="http://www.cameras.com/camera#">
6     xml:base="http://www.cameras.com/camera#">
7
8 <rdfs:Class rdf:about="http://www.cameras.com/camera#Camera">
9   </rdfs:Class>
10  <rdfs:Class
11    rdf:about="http://www.cameras.com/camera#Digital">
12    <rdfs:subClassOf rdf:resource="#Camera"/>
13  </rdfs:Class>
14  <rdfs:Class
15    rdf:about="http://www.cameras.com/camera#PointAndShoot">
16    <rdfs:subClassOf rdf:resource="#Digital"/>
17  </rdfs:Class>
18  <rdf:Property
19    rdf:about="http://www.cameras.com/camera#model">
20    <rdfs:domain rdf:resource="#Camera"/>
21    <rdfs:range
22      rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
23  </rdf:Property>
24  <rdf:Property
25    rdf:about="http://www.cameras.com/camera#effectivePixel">
26    <rdfs:domain rdf:resource="#Digital"/>
27    <rdfs:range
```

```
28         rdf:resource="http://www.cameras.com/camera#MegaPixel"/>
29     </rdf:Property>
30     <rdfs:Datatype
31         rdf:about="http://www.cameras.com/camera#MegaPixel">
32         <rdfs:subClassOf
33             rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
34     </rdfs:Datatype>
35     <model rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
36         Canon PowerShot ELPH 160
37     </model>
38     <effectivePixel
39         rdf:datatype="http://www.cameras.com/camera#MegaPixel">
40         20.0
41     </effectivePixel>
42 </rdf:RDF>
```

Listing 3.1: A description of Canon PowerShot ELPH 16 using RDF Schema.

Similarly, like the above example, any DSLR or Point and Shoot camera can be described using our simple vocabulary and that is what RDFS does.

If an RDF schema is machine-processable, an application may possibly learn some semantics of the property-types mentioned in the schema [5]. The semantics of each of the properties used in a description can be understood by understanding a specific RDF schema. As RDF schemas are constructed on the base of the RDF data model, an application having no understanding of a distinct schema will still be able to parse the description into the property-type and associated values along with the ability to carry the description unscathed.

With this, we come to an end of Section 3.1.1. Of course, this discussion doesn't go too deep into the Resource Description Framework, but it certainly covers the scope of this research so far.

3.1.2 Ontology and OWL

The concept of ontology needs to be clear as it plays a critical role for the Semantic Web. A formal definition of ontology, presented by W3C's OWL Use Cases and Requirements [45] is given below:

“An ontology formally defines a common set of terms that are used to describe and represent a domain. An ontology defines the terms used to describe and represent an area of knowledge.”

This definition makes clear the following issues:

1. It is to be understood that ontology is used to describe and represent an area of knowledge and so is, domain-specific. A specific knowledge or subject area is known as a domain. For example, the area of sports can be considered a domain.
2. There are terms in ontology and these terms have relationships among them. The terms are usually known as classes or concepts. It is to be mentioned that the words, classes and concepts are often interchangeable in regards of 'term'. However, there can be a hierarchical structure in the relationship between the classes. Such as, classes can be divided into super-classes and subclasses. Higher level concepts are served by the super-classes while the subclasses represent finer concepts containing all the higher-concept feature and attributes.
3. Apart from the above mentioned terms, there is a special group of terms which are known as 'properties'. These property terms represent another level of relationships among the classes besides the hierarchical ones. They illustrate diverse features and attributes of the concepts. Properties can also associate different classes together. So it can be said that the classes don't only have the hierarchical relationship of super and sub classes, but also have relationships using properties [6].

So, it should be completely clear by now that we have already stepped into the world of ontology in Section 3.1.1.3. Having defined the terms and relationships of ontology, the basic idea of ontology can be expressed like following: Ontology encodes the knowledge of the domain in such a way that the knowledge can be understood by a computer [6].

3.1.2.1 Benefits of Ontology

As ontology provides a common vocabulary for anyone who needs to share knowledge in a specific domain, it can easily be understood that it one of the main advantages of developing an ontology. Again, ontology allows its users to reuse existing ontologies. While building a large ontology, one can deliberately integrate numerous existing ontologies for describing various parts of the large domain. Moreover, explicit domain specifications make it easy to change the domain assumptions if there is any change or update in a particular domain knowledge. It also helps new users to learn the meaning of the terms used in a domain. Having discussed the mentioned facts, the benefits of ontology can be pointed out as below:

- It shares a universal knowledge or definition among people or software agents about numerous key approaches in a domain,

- enables reuse of existing domain knowledge,

- provides the useful terms for creating RDF documents in the domain,

- makes domain assumptions explicit which enables updating an existing domain knowledge.

Ontology separates domain knowledge from operational knowledge for enhanced user experience in an attempt of reusing existing ontologies and extending them at the same time [46].

It provides knowledge encoding and semantics in machine-understandable format by working together with RDFS, OWL and other ontology description languages.

Finally, it enables domain knowledge analysis once it can access a declarative blueprint of the terms.

3.1.2.2 Web Ontology Language (OWL)

The acronym OWL stands for Web Ontology Language. We introduce OWL in this section by means of a syntax based on RDF. While most of the contents of this section should be accessible without any in-depth knowledge about RDF, the reader may occasionally want to refer to Section 3.1.1.

OWL is a W3C recommended standard for the modeling of ontologies since 2004 [47]. It helps in expressing complex knowledge that RDF(S) fail to represent. For example, let us consider one of the following statements:

1. Every masters student is enrolled to at least one course.
2. The superior of my superior is also my superior.

RDF(S) provides very limited expressive means and that is why it is not possible to represent the above statements using RDF(S). For modeling complex knowledge like these, we need an expressive representation language that works on formal logic and allows logical reasoning. OWL is such a language. It enables access to implicitly modeled knowledge.

In plain English, OWL can be defined as follows [6]:

$$\text{OWL} = \text{RDF Schema} + \text{new constructs for better expressiveness}$$

There are three sublanguages of OWL designed for users to choose from different degrees of expressivity.

These sublanguages are:

1. OWL Lite: less expressive than the other two.
2. OWL DL: contains OWL Lite, decidable, and fully supported by most software tools.
3. OWL Full: only sublanguage that contains all of RDFS, very expressive yet is semantically difficult to understand and to work with. It is hardly supported by any software tools.

Now let us focus on the OWL syntax and semantics. Basically, an OWL ontology is expressed in terms of classes and properties which we have already seen in RDF(S). OWL, however, is able to describe much more complex relationship between classes and properties.

3.1.2.2.1 Header: The header of an OWL ontology contains information about information about namespaces, versioning, and so-called annotations. If required, few more general information about the ontology can be added within an owl:Ontology element. An example of OWL header along with some common information is given in Listing 3.2:


```
1    <rdf:RDF
2      xmlns="http://www.example.org/"
3      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
5      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
6      xmlns:owl="http://www.w3.org/2002/07/owl#">
7
8    <owl:Ontology rdf:about="">
9      <rdfs:comment
10         rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
11         This is an example Ontology.
12       </rdfs:comment>
13       <owl:versionInfo>v0.7.1</owl:versionInfo>
14       <owl:imports rdf:resource="http://www.example.org/foo" />
15       <owl:priorVersion
16         rdf:resource="http://ontoware.org/projects/swrc" />
17     </owl:Ontology>
18 </rdf:RDF>
```

Listing 3.2: The Header of an OWL Ontology.

Some more header elements, such as, `rdfs:comment`, `rdfs:label`, are inherited from RDFS along with `rdfs:seeAlso` and `rdfs:isDefinedBy`.

Apart from the `owl:versionInfo` element in Listing 3.2, the following ones can also be used for versioning:

`owl:priorVersion`

`owl:DeprecatedClass`

`owl:DeprecatedProperty`

`owl:backwardCompatibleWith`

`owl:priorVersion`

Lastly, the `owl:imports` element seen in Listing 3.2 allows importing other OWL on-tologies where the content of the imported ontology is then considered a part of the importing one.

3.1.2.2.2 Classes, Roles, and Individuals: Apart from the basic building blocks (classes and properties) of OWL which we already know by now, there are RDF instances of classes. These instances are called individuals. It is also to be mentioned that properties and roles are often interchangeable as OWL properties are called roles.

Abstract roles and concrete roles are two different type of roles in OWL. Abstract roles connect two or more individuals with each other while concrete roles connect individuals with data values [7].

`owl:ObjectProperty` and `owl:DatatypeProperty` are declarations for abstract role and concrete role, respectively, while both of them are subproperties of `rdf:Property`.

Apart from `xsd:string`, we can also use `xsd:integer` in OWL. All the XML datatypes shown in Figure 3.6 can in principle be used in OWL, but the standard does not require their support.

<code>xsd:string</code>	<code>xsd:boolean</code>	<code>xsd:decimal</code>
<code>xsd:float</code>	<code>xsd:double</code>	<code>xsd:dateTime</code>
<code>xsd:time</code>	<code>xsd:date</code>	<code>xsd:gYearMonth</code>
<code>xsd:gYear</code>	<code>xsd:gMonthDay</code>	<code>xsd:gDay</code>
<code>xsd:gMonth</code>	<code>xsd:hexBinary</code>	<code>xsd:base64Binary</code>
<code>xsd:anyURI</code>	<code>xsd:token</code>	<code>xsd:normalizedString</code>
<code>xsd:language</code>	<code>xsd:NMTOKEN</code>	<code>xsd:positiveInteger</code>
<code>xsd:NCName</code>	<code>xsd:Name</code>	<code>xsd:nonPositiveInteger</code>
<code>xsd:long</code>	<code>xsd:int</code>	<code>xsd:negativeInteger</code>
<code>xsd:short</code>	<code>xsd:byte</code>	<code>xsd:nonNegativeInteger</code>
<code>xsd:unsignedLong</code>	<code>xsd:unsignedInt</code>	<code>xsd:unsignedShort</code>
<code>xsd:unsignedByte</code>	<code>xsd:integer</code>	

Figure 3.6: XML datatypes for OWL [7].

3.1.2.2.3 Class Relations: Just as seen in Section 3.1.1.3, `rdfs:subClassOf` is used for establishing relation between the OWL classes and it is also considered to be transitive as in RDFS. Also, every class is a subclass of `owl:Thing`, and `owl:Nothing` is a subclass of every other class.

`owl:disjointWith` is used for declaring two classes as disjoint. On the other hand, two classes can also be declared as equivalent using the `owl:equivalentClass` element.

3.1.2.2.4 Relation between Individuals: One of the remarkable features of OWL is that it allows declaration of two individuals being in fact the same. `owl:sameAs` allows us to declare this explicitly, but it is also possible that such an identification is implicit, i.e. can be inferred from the knowledge base even without explicit declaration.

`owl:differentFrom` serves the purpose opposite to `owl:sameAs`. Which means that OWL supports declaring that individuals are different. In fact, OWL provides a shortcut to declare that several individuals are mutually different.

3.1.2.2.5 Boolean Class Constructors: OWL allows Boolean relationships between classes, such as intersection or union of two or more classes. It provides `owl:unionOf` for union operation and `owl:intersectionOf` for intersection operation. Certainly, Boolean class constructors can also be used together with `rdfs:subClassOf`. `rdfs:complementOf` is another Boolean class constructor in OWL.

OWL doesn't only consists of the above mentioned syntax and semantics. Rather, it provides us with many more roles and features which allow to build completely user-friendly ontologies. Apart from the above mentioned ones, OWL also allows closed classes, role restrictions and so on.

It is to be mentioned that OWL 2 Web Ontology Language, informally OWL 2, is the latest W3C recommendation for the modeling of ontologies since 2012 [8]. OWL 2 has a very similar architecture to OWL 1. Figure 3.7 shows an overview of the OWL 2 language, showing its main building blocks and how they relate to each other. The ellipse in the center represents the abstract notion of an ontology, which can be thought of either as an abstract structure or as an RDF graph.

At the top are various concrete syntaxes that can be used to serialize and exchange ontologies. At the bottom are the two semantic specifications that define the meaning of OWL 2 ontologies.

OWL 2 adds new functionality with respect to OWL 1. One of the new features is syntactic sugar, such as, disjoint union of classes. There also some other functionality that offer new expressivity. They are:

- keys,
- property chains,

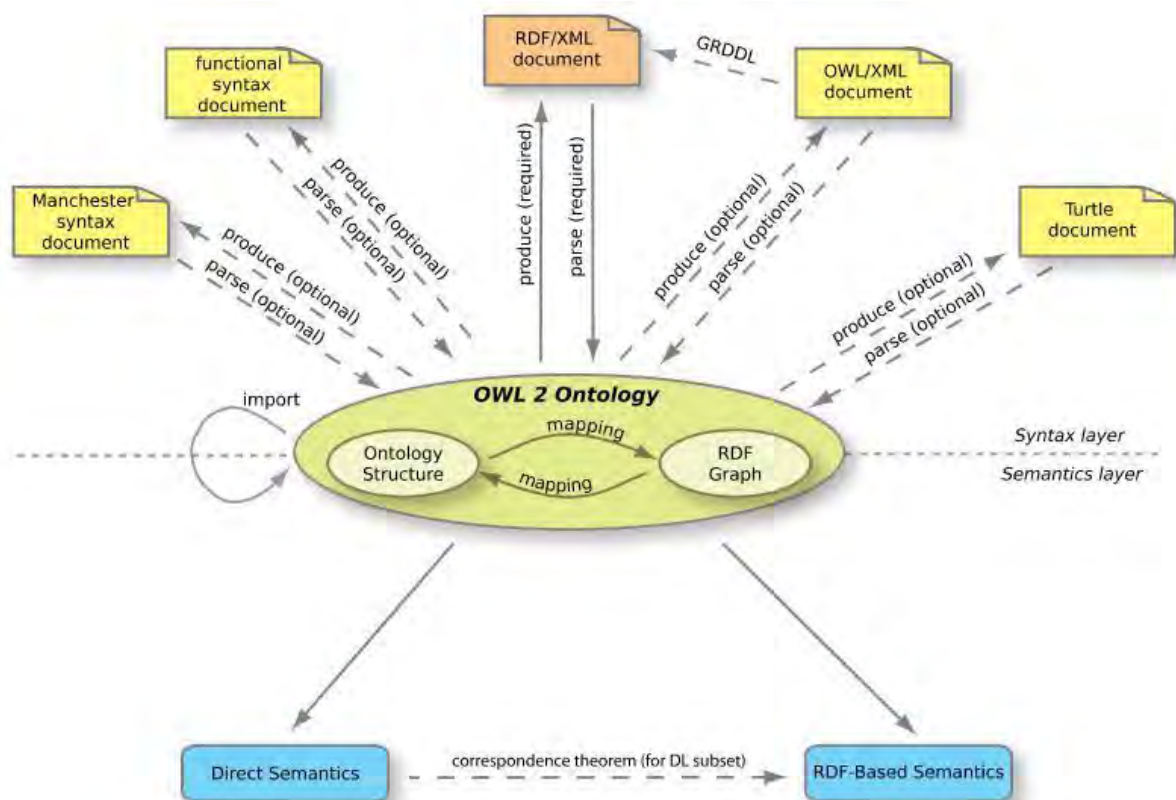


Figure 3.7: The Structure of OWL 2 [8].

- qualified cardinality
- restrictions, richer
- datatypes, data ranges,
- asymmetric, reflexive, and disjoint
- properties; and enhanced annotation
- capabilities.

Not only that, OWL 2 also defines three new profiles [48], namely, OWL 2 EL, OWL 2 QL and OWL 2 RL. These profiles are defined by imposing restrictions on the structure of OWL 2 ontologies. On the contrary, some of the restrictions applicable to OWL DL have been relaxed in OWL 2, which resulted in larger set of RDF Graphs which can be handled by Description Logics reasoners [8].

3.2 Semantic Web Logic and Inferences : Rules at a Glance

Although rule technology has reached notable maturity along with extensive practical usage, it's a crucial challenge to use rules in association with ontologies in Semantic Web. The Semantic

Web has a specific set of goals which vary from most systems of logic. As stated in [49], the following properties are must for a knowledge representation system:

1. A reasonably compact syntax.
2. Adequate expressive power for representing human logic.
3. An efficient, powerful, and understandable reasoning mechanism.
4. A well defined semantics so that one can say precisely what is being represented.
5. Usability for building large knowledge base.

However, a reasoning engine will not be defined by the semantic web itself. It may define valid operations, and may require consistency for them. On the semantic web in general, a party must be able to follow a proof of a theorem but is not expected to generate one [50]. There are various approaches in use which include a set of rule formats such as RuleML, Rule Interchange Format (RIF) [32] etc., and different logic programming such as SWI Prolog that uses RDF data. In this section, we briefly describe two of the semantic web rule approaches, SWRL and SPARQL.

3.2.1 Semantic Web Rule Language (SWRL)

Semantic Web Rule Language allows to use certain kind of rich rules in OWL DL. It is a proposed Semantic Web language combining OWL DL with function-free Horn logic and is written in Unary/Binary Datalog RuleML [32]. A Horn clause can be defined in short as a clause with at most one positive literal belonging to either of the four categories: a rule, a fact or unit, a negated goal, or a null clause. Interested readers are recommended to refer to [51–53] for a detailed view of Horn logic.

As mentioned earlier, SWRL allows Horn-like rules to be used in association with OWL DL ontologies and includes a high-level abstract syntax. The SWRL rules are saved as part of ontology. All rules in SWRL are expressed in terms of OWL concepts: classes, properties, individuals. An example SWRL rule of assigning property value is given below:

$$\text{Person}(?p) \wedge \text{hasSibling}(?p, ?s) \wedge \text{Man}(?s) \rightarrow \text{hasBrother}(?p, ?s). \quad (3.1)$$

This rule depicts that a Person p , having a sibling s , who is a Man, can be assigned with the property hasBrother which means that s is brother of p . Some of these rules can not



Figure 3.8: SWRLTab Built-In Libraries [9].

be defined in OWL 1.0 whereas SWRL has built-ins that dramatically increase expressivity [9]. Such built-ins of SWRL are `swrlb:greaterThan` and `swrlb:lessThan`. For example, any person having age between one month and one year can be referred to as ‘infant’. SWRL expresses this piece of information as follows:

$$\text{Person}(?p) \wedge \text{hasAge}(?p, ?age) \wedge \text{swrlb:greaterThan}(?age, 0.08) \wedge \text{swrlb:lessThan}(?age, 1) \rightarrow \text{Infant}(?p, ?s). \quad (3.2)$$

SWRL can also define new built-in libraries including mathematical built-ins such as `swrlm:eval`. We can see the SWRLTab built-in libraries in Protégé editor in Figure 3.8.

Eventually, SWRL lies at the other end of the integration of description logic and function-free rules while compared to OWL2 RL [54].

3.2.2 SPARQL

SPARQL (pronounced as ‘sparkle’) is an RDF query language. It is a semantic query language for databases which is able to retrieve and manipulate data stored in RDF format. SPARQL has capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. It also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph.

The basic syntax of SPARQL SELECT Query is as follows:

```
1 SELECT ?subject ?object
```

```
2 WHERE { ?subject rdfs:subClassOf ?object }
```

Rules can be expressed in SPARQL using its CONSTRUCT feature. For example, the following rule, $brother(X, Y) \leftarrow brother(X, Z), brother(Z, Y)$ can be expressed in SPARQL as:

```
1 CONSTRUCT {  
2     ?X brother ?Y.  
3 } WHERE {  
4     ?X brother ?Z.  
5     ?Z brother ?Y.  
6 }
```

The above brief discussion covers the scope of our research. At the end, it should be kept in mind that SPARQL is a rule language and NOT an implemented rule system and so is able to express one inference step [32].

With this, we end this chapter gathering basic knowledge of Semantic Web and the related technologies. We will now proceed to the detail of our proposed model from Chapter 4.

Chapter 4

Proposed Framework

In this chapter, we propose a framework which can be used in any online community. But in our research, as a specific case, we intend to deal with individual's National Identity or NID information as a case study of online community. At first, we will discuss the basic terminologies related to our work. Then, we will go into the detailed view of our proposed framework.

4.1 Definitions and Brief Overview

4.1.1 National Identity (NID):

As mentioned earlier, a National ID Card is a portable document for confirming an individual's national identity. It is typically a plasticized card with information encapsulated digitally. In Bangladesh, the National Identity Card (NID Card in short), is a mandatory document that is issued to every Bangladeshi citizen upon turning 18 years of age. The card holders currently receive 22 types of services, which include banking, TIN, driving license and passport [55]. Apart from the mentioned information, the Bangladesh Election Commission is planning to include the following facilities in the enhanced version of citizens' NID cards, known as smart cards:

- Citizens Right and Benefits, National Identity,
- Driving License,

- Passport,

- Property Land Buying and
Selling, Open Bangladeshi Bank
Accounts, Bank Loans Support,

- Government
Allowances, Support
Received,

- BIN Facility,

- Share-BO Account
Maintainers, Business Trade
License,

- Vehicle
Registration,
Insurance
Schemes,
Marriage
Registration, E-
passports,

- E-Governance,

- Gas and Electricity

Connections, Mobile

Connect,

- Health

Cards, E-

Cash,

- Bank Transactions,

- Students' Admission Facilities [56].



Figure 4.1: Format of NID Card.



Figure 4.2: Sample Page for NID Network.

A sample view of the current NID card format of Bangladesh is shown in Figure 4.1.

It is to be mentioned that not only in Bangladesh, but around 100 countries had enacted laws making identity cards compulsory according to a 1996 publication by Privacy International [57]. In these countries, the card must be shown on demand by authorized personnel under specified circumstances [58].

4.1.2 NID Network

As the information mentioned above is also stored in database, so the citizens may have the opportunity to retrieve their data from this database. That is why an online network is assumed for NID holders for the purpose of this research. Referred to as 'NID Network', users can log into the community (Figure 4.2) like any other OSN and have control over the sharing of their information.

Although there is no known existing online community as the one proposed in this research, countries like United Kingdom, U. S., India have databases for their citizens holding national

identity card. Among them, the United States has developed the most decentralized and re-stricted system for permitting citizen access [59]. In this system, users may query government databases only and must submit a written request directly to each agency that collects personal information. But the necessity for access to personal information does not end at government services. An inability to access, verify, update, and delete records in private databases may equally infringe citizens' privacy. In most cases, the United States is resembled by the United Kingdom. While these countries are highly emphasizing on the security and privacy of their citizen information, recently India's national ID database is reported to be accessed for less than USD 10 [60]. Reports from the country suggest that the government's national ID system — Aadhaar, holding personal data belonging to more than one billion people — was compromised. It is to be mentioned that Aadhaar isn't compulsory in India. But compromise in such confidential database is really an issue requiring high concern. That is why the security of such information database must be ensured.

Though the 'NID Network' will not function completely as an OSN, it has some basic features identical to OSN. Such as, uploading the identification photo of a citizen, updating own information on condition of verification, connecting with other NID holders etc. Unlike traditional OSN, as there is no option like 'wall sharing' or any posting message area in our discussed network, wall filtering [22] won't be of any use for our proposed methodology. Rather, in an NID Network, there would be fields of data or information. As the information present in the NID database are very secured and sensitive, we won't be using actual data of citizens. Instead we'll use synthetic realistic data for testing purpose.

4.2 Detailed Framework

In an NID Network, though core item of the network is an individual's NID, other personal information, e.g., financial and medical data, can definitely become integrated to the network.

Let us assume that a bank launches an online KYC (Know Your Customer) application that automatically verifies the basic client data from a user's NID profile present in the NID network. When the relevant user grants permission for application installation, it is automatically given access only to the content required for account opening and nothing else. As such, the application will not be able to view or access the user's medical content. This process on one hand, saves time in verifying a client's information. On the other hand, access to irrelevant contents

can be prohibited.

Whatever the network be, OSN or NID, an access control system should have the elements (adapted from [24]) delineated in the following sections.

4.2.1 User Profiles

Every user in our context, i.e., NID holder has a user profile which is a list of identification information, such as name, sex, date of birth etc. For an NID holder, the profile may be identified by his/her unique NID number. In Bangladesh, a citizen is assigned a 13 or 17-digit identification number as NID number. In traditional OSNs like Facebook, users are also given a unique identifier using which their profiles are identified. Similarly, the NID Number can also be used for identifying a citizen's profile in the NID Network. While there is no clear theory regarding how Facebook IDs got assigned once Facebook expanded beyond college networks [61], National ID Number has its own meaning for each citizen [62].

4.2.2 Content

Contents in online networks may vary in different communities; including the user profile information in Section 4.2.1, photo images, blog entries and so on.

Currently, the Election Commission Bangladesh has a portal for voter information of all the NID holder [63]. Citizens can log into the portal using their NID Number or the Form number (last six digits of NID Number) and see their voter information (Figure 4.3).

In our NID Network, all content will be the information present in the database against the user's NID. We can categorize these contents as follows:

4.2.2.1 Submitted Content

For obtaining the NID, a citizen must submit a series of information, i.e., contents to the govt.

These contents can be of different classification, such as:

4.2.2.1.1 Basic Identifying Content: This category consists of the elemental information that are needed to identify a citizen uniquely. All the information embedded in the NID Number are included in this category. Additionally, the user's photograph and signature are also considered to be in this group.

Figure 4.3: NID Service of Election Commission Bangladesh.

4.2.2.1.2 Biometric Personal Content: As the Govt. of Bangladesh has started replacing the traditional NID Card with ‘Smart NID Cards’, physiological and biological characteristics of the individual are collected that make it easier to identify that individual with increased accuracy and less error. For example, fingerprint, face recognition, iris recognition etc. are some of the most popular biometric identification process. Smart NID Cards require impression of all ten fingers of the citizens along with iris recognition. Each one of this information is must for the Smart Card holders.

4.2.2.2 Uploaded Content

Besides the required content, the users may upload various contents in the NID Network, just like they do in OSNs. As the enhanced NID Cards plan to include several facilities like health facilities, transaction facilities, etc., so the users may publish other contents according to the requirement of these privileges against their NID. For example, if s/he has any health issues, which doctor(s) s/he visits, and the prescriptions written; or else, which bank preserves accounts attached with her, what are the banking services she avails etc.

4.2.3 Privacy Settings

A user's privacy setting represents his/her requirement to share contents with each of his/her friends or other users. In NID Network, two users having any type of relationship with each other will be considered to be in each other's loop. The term loop, thus, will be used to represent a set of users having relationship with each other in NID Network. Now, let us assume that a user has a list of some other users, say L , in her loop and let C be her content. The user's privacy settings can be viewed as a $|L| \times |C|$ matrix, where each entry is a policy specification defined by her. In NID Network, the settings will be dependent on content types. If the content is labeled as 'Sensitive', its default setting will be the access right available only to the owner. If it's marked as not sensitive or 'Common', it will be accessible to all in the loop. Of course, the user can change both the settings, as well as make it accessible to certain users, as already mentioned beforehand. Privacy settings are discussed in detail in Chapter 5.

4.2.4 Ontology APIs

For a semantically developed system, some APIs need to be integrated with the traditional system. This includes two knowledge bases:

4.2.4.1 Online Network Ontology (ONO)

For capturing the information semantics in a virtual community. One of the most popular ontologies for building an online community is the FOAF (Friend Of A Friend) Ontology started in the year 2000 [64]. FOAF uses the RDF/XML or RDFa syntax for publishing FOAF descriptions as linked documents in the web. As we tend to model the NID Network using the Semantic Web technologies, Online Network Ontologies like FOAF may certainly come handy in building our target community. Such as, for representing that Bank A and Bank B are members of a group 'Banks', we can write the following lines of codes shown in Listing 4.1.

```

1 <?xml version="1.0"?>
2   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:dc="http://purl.org/dc/elements/1.1/">
4     <foaf:Group xmlns:foaf="http://xmlns.com/foaf/0.1/">
5       <foaf:name>Banks</foaf:name>
6       <foaf:member>
7         <foaf:organization>

```

```

8           <foaf:name>Bank A</foaf:name>
9           </foaf:organization>
10          </foaf:member>
11          <foaf:member>
12          <foaf:organization>
13          <foaf:name>Bank B</foaf:name>
14          </foaf:organization>
15          </foaf:member>
16          </foaf:Group>
17         </rdf:RDF>

```

Listing 4.1: An example of Online Network Ontology (FOAF).

4.2.4.2 An Access Control Ontology (ACO)

This is used to present and store any knowledge entirely needed for the purpose of access control. It also includes inferences based on access control rules or policies. We present an ACO for the proposed model in Section 7.2, that depicts the control based on content type.

4.2.5 Access Control Rules or Policies

Each user in an online community requires a set of rules and related policies in order to control her content sharing. So does the NID Network. In fact, it is the most important prerequisite for the NID Network as all the content present in the network must be genuine.

Let us suppose that an NID user wants her current prescription to be viewed only by her doctor. Then she will be able to set a policy for that content which may look like below:

<PrescriptionX, Sensitive, Viewed only by Dr. X>

In our proposed model, there will be two parties for setting up/controlling the access control rules and policies. First party will be the one who owns a specific content while the latter one will have the right to regulate the existing policies to a limited extent only if allowed by the content owner. The details of these policy specifications for the NID Network are discussed in Chapter 7.

With this, we finish describing the framework we propose for the NID Network. So the outer structure of the proposed model is ready. We will proceed to the next chapter discussing the methodologies needed to deploy the framework.

Chapter 5

Methodology Details

Having discussed our framework in the previous chapter, we need to put forth the methodology for actual access control. The proposed model can be referred to as the Content-Based Access Control model or CBAC model in short. It is inspired by MPAC model [4], and thus, some of the entities mentioned in [4] have been reused here. But unlike MPAC, it focuses mainly on content type instead of user relationships. Not only that, this model also deals with sticky policies, privacy settings of contents and their co-privacy. This methodology presents a reusable ontology for the proposed model as well. The shared contents may be connected to multiple users and thus have stakeholders.

In our work, a new attribute, *CnT*, is introduced for defining whether the user data is sensitive or not and later on define a rule for sharing that content using the attribute. Here, the entities involved can be divided into three major classes:

1. Entities dealing with the citizens.
2. Entities related to the NID attributes (contents).
3. Relationships dealing with the interconnection between the members of the aforementioned two.

The following sections delineates the elaborations of the above.

5.1 Citizens

In the NID network, only the citizens having NID can be present. All the contents prevalent in the network must be possessed by an individual having an NID. So, in the proposed CBAC model, Possessor is the main controller of a content:

Definition 5.1.1 (Possessor). Let c be a content in the space of an NID holder, z ($z \in Z$, as detailed below) in the NID network. Here, z is called the possessor of c .

Z is a set of users in the online community (for us, the NID Network). As NID is an individual's identity of citizenship of a country, each user in the NID Network is a citizen of that nation. The set can be declared as, $Z = \{z_1, \dots, z_n\}$. Each citizen has a unique identifier. In the NID Network, the NID number for each citizen plays the role of the unique identifier. Each citizen in the network has a profile containing his/her necessary attributes. For example, a citizen's NID number is an attribute of that particular citizen. Each attribute in profile is entered as a attribute-value pair, $q_{ij} = \langle \text{attr}_i : \text{pvalue}_j \rangle$, where attr_i is an attribute identifier and pvalue_j is the attribute value. A citizen's owning an NID Number can be expressed as: $q_{ij} = \langle \text{NIDNo} : \text{Yes} \rangle$.

As we have already mentioned, in order to include other citizens in a group, we use loops. While adding each other in a loop, individuals may want to classify other citizens in identifying groups or clusters which we denote by knot. So, our next entity can be,

$Kn = \{kn_1, \dots, kn_n\}$, which is a set of knots to which the users can belong.

Let us make the idea of knot a bit clearer at this point. Suppose, citizen A has a current account in Bank C, a savings account in Bank D and a fixed deposit in Bank E. A can group all three of these banks under a knot named Bank.

Putting somebody or some other entity into a knot is optional for citizens. But there are cases when users are added to knots automatically depending on some conditions. Such a knot is the Parents knot. When a citizen is marked as father or mother of another citizen, (s)he is by default added to the Parents knot. An example will give a clearer view. Let us suppose that the citizen A has Y and Z in her loop. As per A 's NID information, Y and Z are her mother and father, respectively. So both Y and Z are automatically added to the knot, Parents.

A citizen will have a set of relationships with other citizens in his/her loop. So, let us define this as follows:

$R = \{r_{z1}, \dots, r_{zn}\}$ is a collection of user relationship sets. Here, each member in the collection is a set of relationships of the corresponding user. Thus, $r_{zi} = \{s_{zi1}, \dots, s_{zim}\}$ is the relationship list with m citizens of a citizen $z_i \in Z$.

Each relationship entry is a citizen : relationship-type pair, $s_{ij} = z_i : rt_j$, indicating relationships of citizen z_i with citizen z_j , where $(z_i, z_j) \in Z$, $rt_j \in RT$.

Let us define RT now.

RT is a set of relationship types supported by the NID Network. Each user in the community may be connected with others by relationships of different types. The supported relationships are: Father, Mother, Child, Sister, Brother, Cousin, Friend, Patient, Doctor, Client, Vendor, AccountHolder and similar ones.

To depict that Y is Mother of A , it can be written as: $s_{AY} = \langle A : \text{Mother} \rangle$. Again, a mapping function can be used to map the citizens in specific relationships in a citizen's loop.

loopMembers is a function mapping each citizen $z_i \in Z$ to a set of citizens with whom s/he has a specific relationship $rt \in RT$. Thus, these members are actually in his/her loop. As such, $\text{loopMembers}(z_i \in Z, rt \in RT) = \{z_j \in Z \mid (z_i, z_j) \in s_{ij}\}$.

Let us suppose that citizen A works in the Marketing department of a Finance company and so has to manage communication with various potential clients. For verifying the genuineness of the client, A adds her clients to her loop in the NID Network. X is one of the clients of A . As NID Network supports the Client relationship, so X can be defined by A as Client. Now, using the mapping function loopMembers, citizen A having a relationship Client with X can be depicted as: $\text{loopMembers}(A, \text{Client}) = \{X \in Z \mid (A, X) \in s_{AX}\}$. Here, $s_{AX} = \langle A : \text{Client} \rangle$. All the citizens having a relationship Client with A will belong to this loop.

5.2 NID Attributes (Contents)

As already mentioned, an NID belonging to a particular user will have a number of attributes, which include NID number, profile photo, date of birth, signature and so on. We call these attributes contents of a citizen in NID network.

Definition 5.2.1 (Content Specification). Let $c \in C$ be a content in the space of a citizen z in the NID network. The content specification is defined as a tuple: $\langle Z, e_{zj} : cnt \in CnT \rangle$.

The entities involved in the above definition are described below:

$C = \{c_{z1}, \dots, c_{zn}\}$ is a collection of user content sets, where c_{zi} is a set of contents of a citizen $z_i \in Z$. $CnT = \{\text{Sensitive}, \text{Common}\}$ is a set of Content Type which identifies if a content is sensitive to share with others or not. As mentioned earlier, a citizen may, of course, want his/ her NID attributes to be secured from being misused. So he/she may want to define an attribute as Sensitive or Common. As CBAC deals with NID attributes, i.e., user contents, this purpose is served by introducing CnT defining types of the attributes.

Each content is a pair of $\langle \text{content identifier} : \text{content type} \rangle$ pair, $\langle e_{zig} : cnt \in CnT \rangle$, where e_{zig} is a particular content identifier of citizen z_i . User A has contents c_{A1} , c_{A2} and c_{A3} in her content sets which are her date of birth (dob), profile photo (pp) and signature (sig) respectively. A wants her date of birth (dob) to be visible to all while she doesn't want the remaining two contents to share with anyone. So the pairs specifying her contents mentioned above look like these:

$c_{A1} = \langle e_{Adob} : \text{Common} \rangle$ for the first content, date of birth (dob). The content has a unique identifier e_{Adob} which reflects that this specific date of birth belongs to citizen

A . As A wants to share her date of birth publicly, she sets the attribute $cnt \in CnT$ as Common. Which is why the content c_{A1} shows the setting for c_{A1} to be Common.

$c_{A2} = \langle e_{App} : \text{Sensitive} \rangle$. This pair shows the setting for A 's next content, profile photo (pp), which she doesn't want to share with anyone but herself. So the content-identifier for this content pairs with the content type Sensitive.

$c_{A3} = \langle e_{A\text{sig}} : \text{Sensitive} \rangle$. A definitely doesn't want her signature to be visible to anyone as this piece of content is actually sensitive to any user. So, A 's third content, signature (sig) also has the content type Sensitive like the previous one.

Further elaboration on this can be found in Chapter 7.

5.3 Interconnection between Citizen and Content

As owner of an NID, a citizen will have the ultimate control of his/her NID attributes. But there will be other citizens also playing the role of stakeholders of a content. These citizens, naturally, will not have the same control as the owner over the content. Rather, there will be limited control for them defined by the owner. For example, in order to verify the address of an account holder, a bank may want to see the content piece containing address of citizen A in the NID Network. As the owner, A has full control over this content and rights to define access control for the mentioned bank. So, there must be some interconnecting elements between contents and the related citizens. In order to deal with this issue, we use content controllers.

CT is a set of content controller types. A content can have two types of controllers. One who owns or possesses it, another who can view or access it. So, the controller type set can be declared as, $CT = \{P\ S, AS\}$ where PS and AS indicates possessorOf and associateOf, respectively. Throwing back to the recently mentioned example of citizen A and her bank, it can be understood that A is the PS of her address as she is the owner or possessor, while her bank may be an associate (AS) of this content defined by A . Details of these controllers are discussed in Chapter 6.

Thus we have reached the end of this chapter defining the required entities and connections needed to be established between them. So we can now proceed to a more detailed view of the model discussing the issue of control over content sharing, which is the goal of this research.

Chapter 6

Content Sharing Controls

As discussed in Chapter 5, we now have defined all the items which are necessary for the proposed model. In this chapter, we'll focus on the sharing of user content based on the type and determine the entities associated with content.

6.1 Content Sharing in OSN

Traditional OSNs provide built-in mechanisms for users to communicate and share contents with each other. The users are allowed to post both text and image contents in own and others' spaces, mention each other in the posts and share these contents. So each content definitely has an owner and may have other users, or stakeholders, possessing some controls over it.

6.1.1 Content Sharing from Own Space

The very basic facility of traditional OSNs is that they allow users to publish content in their own space with different sharing privileges. These contents can be simple text or image or both. They have control over who can view their content and who can't. Users can select some specific individuals who can see what they have published while they can also make the shared content visible for every user in the OSN space.

As can be seen in Figure 6.1(a), the user has shared her content with only the users in her friend list which means the content is visible to only the users who has a friend relationship with her.



(a) Content sharing with friends only.

(b) Content sharing with all users in OSN space.



(c) Content sharing with specific individuals.

Figure 6.1: Example of Content Sharing in Social Network like Facebook.

On the other hand, the job post in Figure 6.1(b) is shared publicly so that every Facebook user can view it. Figure 6.1(c) shows that the content is shared only with some people in the user's space and except them, even no other friend can view it.

In all cases, no other user than the owner can control the content or its sharing. Only the user who possesses the content has full control over it. Neither can he/she give the control to any other user.

6.1.2 Content Sharing in Others' Spaces

Traditional OSNs also give users option to share content in other users' spaces. It can be done in two ways. Either the users can use the 'share' button below a content or they can browse to another user's space and publish content there directly.

In Figure 6.2(a), the general content sharing in others' space using the 'share' button shows four options to share a content:

- In user's own space,
- In some other individual's space,



Figure 6.2: Example of Content Sharing in Other User's Space.



Figure 6.3: Example of Sharing Other's Content in Facebook.

- In a group of users,
- On a page the user manages.

On contrary, Figure 6.2(b) shows that a user has directly published something in another user's space. For both of these, the users need to keep the facility of publishing content in their space open for other users. The control over viewing the content depends on the user at whose space it is shared. S/he may allow no one to see the contents shared on her/his space or select some specific people to view them.

6.1.3 Sharing Other Users' Contents

This one is quite similar to Section 6.1.2. Although a user's content can be shared by other users using the 'share' button as seen in Figure 6.3, it can't be modified by anyone except the owner. The sharing user can only control the viewing facility in his/ her own space.

6.1.4 Tagging Users in Content

Apart from the three types of content sharing described till now, traditional OSNs allow users to mention each other in their published contents. This method of mentioning is called 'tagging'



(a) Mentioning users in text content.

(b) Tagging users in photo content.

Figure 6.4: Example of Tagging in Social Network.

in the field of social networks. Users can mention each other in their own as well as other users' contents if permitted. The mentioned user is called the 'tagged user'.

Figure 6.4(a) shows a user content with other users mentioned in it while Figure 6.4(b) allows users to select other users for mentioning in a photo content. In both cases, the tagged user has the authority to control the viewing of the tagged content in own space. S/he can allow the content to be visible in own space or generate a 'Remove tag' request if the content is a photo.

6.2 Content Sharing in NID Network

As mentioned earlier, sharing a content in the NID Network is quite different from content sharing in social networks. Chapter 4 defines the main structure of the NID Network with all its elements. For content sharing, we need to define some more components.

6.2.1 Required Definitions

Besides the owner of a data item or content, there are those who are associated (view or regulate or both, if permitted) with it. These sets of citizens can be defined as below:

Definition 6.2.1 (Associate). Let c be a content in the space of a citizen $z_i \in Z$ in the NID network, where z_i is the Possessor of c . Let V and W are the sets of citizens associated with c . V and W are the set of citizens with whom A has decided to grant view only permission and regulate permission respectively for c . Definitely, $V \subseteq Z$, $W \subseteq Z$, $V \cup W = Z$. A citizen z_j is called an associate of c , if $z_j \in V$ or if $z_j \in W$. As we have already mentioned above, it is understandable that associate permission can be of two types: view only and regulate.

Definition 6.2.1.1 (Viewer). A Viewer is a citizen who can view a content if allowed, but can not regulate the settings of it.

Definition 6.2.1.2 (Regulator). A Regulator is a citizen, who, along with the Possessor, can control the settings of a content or data item.

While defining access control for content, Possessor, Viewer and Regulator, each can be viewed as an individual controlling the content.

A citizen will have a set of associates. The specification of associates of a particular citizen will express the type of permission the associates will have on the contents owned (or controlled) by that citizen. The associate specification is defined as a set, $associate = \{as_1, \dots, as_n\}$. Here, each as_i is a tuple, $\langle I, PS/AS, pt \rangle$, where, I = an individual citizen, loop members or a knot, PS/AS = type of associate, PS or AS, pt = permission type: View Only or Regulate.

Let us delineate a few examples to clarify the above. Suppose that citizen B is in A 's loop and has View Only permission over A 's date of birth. So the associate specification for B will be like: $\langle B, AS, View Only \rangle$ Here B represents one of A 's loop members, AS is the type of association of B and View Only is the type of permission granted to B .

Again, let us suppose that A has a knot named Banks consisting the banks in her loop. She wants this knot to view her NID No for verifying purpose of her account(s) in these banks. So, similarly, the associate specification for the Banks knot can be written as: $\langle Banks, AS, View Only \rangle$

Another example can be the Regulate permission of A 's parents over any of her contents,

such as her profile photo, when she is an underage. The association specification $\langle Parents, AS, Regulate \rangle$ expresses the Parents knot in A 's loop while AS and Regulate are the association type and per-mission type of that specific knot, respectively.

Based on the definitions and rules above, the general sharing rule for a content can be specified as: for a citizen z_i , with an age z_{age} and possessing a content c_1 will be a tuple $(z_i \in Z, z_{age}, c_1 \in C)$. The previous tuple can be elaborated as, $(z_i, c_1, PS) \wedge (z_j \in Z) \implies (z_j \in loopMembers(z_j, rt) \wedge (z_j, c_1, AS))$. This implies that another citizen z_j who is connected to z_i with a relationship type rt and thus is a loop member of z_i , is also an associate (AS) regarding content c_1 , which is granted by z_i .

Age of the user is required here because in our proposed model, there will be an age limit for each citizen. If the citizen's age is less than the limit, her parents will automatically be given

access to her contents and control over the sharing. It will be done by an intersection operation of both the user and her parents' settings and this common setting will be applied on the content. If any Possessor or Regulator declares a content to be of Sensitive type, the type will be set as Sensitive whatever setting it might previously have.

With this, the elements needed for content sharing in the NID Network finish getting defined. Based on these definitions, access control policies will be defined in next chapter.

Chapter 7

Access Control Policies

This chapter outlines the controls placed on the shared information in the NID Network. For this, we first define the required policies as adequate security of user contents is a basic management responsibility. Next, using the benefit of universal knowledge sharing, we have developed an ontology that implies with the defined policies in Section 7.1.

7.1 Policy Specification

Here we define the policies of our proposed model. These policies express the access control rules stated above for a content in meaningful formats. Possessors can set policies for their content while Regulators also have the right to control the policies. Based on Definition 5.2.1, the CBAC policy for content sharing will be a 4-tuple as mentioned below:

$P = \langle \text{possessor}, \text{citizen_age}, \text{content}, \text{shared_with} \rangle$, where,

1. $\text{possessor} = (z \in Z)$ is a citizen who possesses a content, which means she has the full control to regulate the access of content.
2. citizen_age represents the age of the citizen.
3. $(\text{content} \in C)$ represents the published information along with its sensitivity. As mentioned earlier, each attribute of the NID can be considered as a content.
4. shared_with is a set of citizens who can view or regulate the published content. In a word, shared_with is a set of associates of the content.

Let us suppose that citizen A defines her “date of birth” ($A.dob$) as Sensitive and does not select anyone from her loop to share with. Again, she wants to share her NID number ($A.NIDNo$) with her whole loop but allows no one with Regulate permission and allows her parents with Regulate permission to her address while restrict both View and Regulate for all other users. When a user defines any information as Sensitive, by default it is shared with only herself. So the policies for above state conditions for A will be:

$$p_1 = (A, 20, \langle A, e_{A.dob} : \text{Sensitive} \rangle, \langle A, PS, \text{Regulate} \rangle). \quad (7.1)$$

This 4-tuple policy contains all the information needed to define control over the content $e_{A.dob}$ of user A . The first element is a tuple expressing A 's controller privilege over $e_{A.dob}$. That is, A is the Possessor of content $e_{A.dob}$. Next element states A 's age ($A.age$), then comes the content specification as per Definition 5.2.1. This element holds in a tuple A 's profile, her relationships and the content in concern with its type: Sensitive. As she hasn't yet given association to anyone else in her loop, the default privacy setting of $e_{A.dob}$ is set to own self, i.e., only A has the permission to view and regulate her own date of birth.

$$p_2 = (A, 20, \langle A, e_{A.NIDNo} : \text{Common} \rangle, \langle A, PS, \text{Regulate} \rangle, \langle *, AS, \text{View Only} \rangle). \quad (7.2)$$

Here p_2 , like p_1 (Equation (7.1)), states Possession of A over her another content NID No ($A.NIDNo$) and her age ($A.age$). It also represents whether A 's NID No should be visible to all or not. Here, $e_{A.NIDNo}$ content is Common. Finally, as A has decided to share her NID No with all her loopMembers, the last element shows the tuple containing this setting. The asterisk in the tuple means all users in A 's loop having any relationship with her. This last tuple also sets a View Only permission over A 's loop over content $e_{A.NIDNo}$.

$$p_3 = (A, 20, \langle A, e_{A.address} : \text{Sensitive} \rangle, \langle A, PS, \text{Regulate} \rangle, \langle \text{Parents}, AS, \text{Regulate} \rangle). \quad (7.3)$$

This policy expresses privacy setting for A 's third content, her Address($A.address$), which is marked as Sensitive by A . This content is shared with only A 's father and mother as per A 's choice. A 's father (X) and mother (Y) are already clustered as Parents in NID Network, the last element of p_3 's 4-tuple policy states Regulate permission for the Parents knot.

Now let us suppose that the Age Limit in the system is 25 years and so A can be considered as Underage because A has age of 20. That's why, X and Y will have Regulate permission to

A 's contents by default. So p_1 , p_2 and p_3 (Equations (7.1) to (7.3)) can be rewritten as below:

$$p_1' = (A, 20, \langle A, e_{A.dob} : \text{Sensitive} \rangle, \langle A, P S, \text{Regulate} \rangle, \langle \text{Parents}, AS, \text{Regulate} \rangle). \quad (7.4)$$

$$p_2' = (A, 20, \langle A, e_{A.NIDNo} : \text{Common} \rangle, \langle A, P S, \text{Regulate} \rangle, \langle *, AS, \text{View Only} \rangle, \langle \text{Parents}, AS, \text{Regulate} \rangle). \quad (7.5)$$

$$p_3' = (A, 20, \langle A, e_{A.address} : \text{Sensitive} \rangle, \langle A, P S, \text{Regulate} \rangle, \langle \text{Parents}, AS, \text{Regulate} \rangle). \quad (7.6)$$

Having been rewritten, p_1' , p_2' and p_3' (Equations (7.4) to (7.6)) now set Regulate permission over all of A 's contents by default to her Parents group.

Under a different scenario, let us say that A 's parents (X and Y) do not want her NID No to be visible to all but some specific citizens who are within all of their mutual contacts. While A sets her address as Common and allows all her loop members to view it, X suggests that it should be visible to P , Q and R who are in his loop and Y suggests it to share with M , N , P , R and Z from her loop. Now, as A has made her address visible to all and let us assume that she has M and P in her circle, intersecting all three settings from A , X and Y , for A 's address, the content type will become Sensitive and it will be visible to M and P only. So, p_2 is re-written:

$$p_2'' = (A, 20, \langle A, e_{A.NIDNo} : \text{Common} \rangle, \langle A, P S, \text{Regulate} \rangle, \langle M, AS, \text{View Only} \rangle, \langle P, AS, \text{View Only} \rangle, \langle \text{Parents}, AS, \text{Regulate} \rangle). \quad (7.7)$$

Thus citizens may set access control policies for any other content in their spaces. These are the settings for contents in a citizen's own space. This is not the same with tagged contents, i.e., contents with other citizens mentioned with them. Privacy settings for tagged contents are called co-privacy of a content. Co-privacy settings are discussed later on in Chapter 9.

7.2 Ontology

To properly implement the policies, we have developed an ontology for our framework proposed in Chapter 4. In this ontology, we have presented the sets mentioned previously using classes. Currently the main classes in the ontology are Citizen, Content, ContentType and RelationType. Some relevant classes to the mentioned classes are also present as subclasses in the CBAC

Ontology. The super class of all the classes is NIDNetwork, under which all other classes have been declared.

Let us first discuss the Citizen class. This class contains all the users having an NID present in the NID network. It has four (overlapping) subclasses, which are: Possessor, Associate, Knot and UnderAge. The OWL definitions of the Citizen class and its subclasses are shown through Listing 7.1 to 7.5.

```
221 <owl:Class rdf:about="http://www.semanticweb.org/ontologies
    /2016/11/cbaontology#Citizen">
222 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
    ontologies/2016/11/cbaontology#NIDNetwork"/>
223 </owl:Class>
```

Listing 7.1: Definition of Citizen class in CBAC Ontology.

```
203 <owl:Class rdf:about="http://www.semanticweb.org/ontologies
    /2016/11/cbaontology#Associate">
204 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
    ontologies/2016/11/cbaontology#Citizen"/>
205 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
    ontologies/2016/11/cbaontology#NIDNetwork"/>
206 </owl:Class>
```

Listing 7.2: Definition of Associate class in CBAC Ontology.

```
271 <owl:Class rdf:about="http://www.semanticweb.org/ontologies
    /2016/11/cbaontology#Possessor">
272 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
    ontologies/2016/11/cbaontology#Citizen"/>
273 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
    ontologies/2016/11/cbaontology#NIDNetwork"/>
274 </owl:Class>
```

Listing 7.3: Definition of Possessor class in CBAC Ontology.

```

246 <owl:Class rdf:about="http://www.semanticweb.org/ontologies
      /2016/11/cbaontology#Knot">
247 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
      ontologies/2016/11/cbaontology#Citizen"/>
248 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
      ontologies/2016/11/cbaontology#NIDNetwork"/>
249 </owl:Class>

```

Listing 7.4: Definition of Knot class in CBAC Ontology.

```

298 <owl:Class rdf:about="http://www.semanticweb.org/ontologies
      /2016/11/cbaontology#UnderAge">
299 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
      ontologies/2016/11/cbaontology#Citizen"/>
300 <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/
      ontologies/2016/11/cbaontology#NIDNetwork"/>
301 </owl:Class>

```

Listing 7.5: Definition of UnderAge class in CBAC Ontology.

As mentioned earlier, the citizen who owns or possesses a content belongs to the Possessor class. Similarly, Associate and Knot classes have purposes as mentioned previously.

As CBAC allows Supervisory Content Settings for parents, we declare the subclass UnderAge within Citizen class. This class is defined using the following SWRL rule:

$$\text{Citizen}(?x) \wedge \text{citizenAge}(?x, ?a) \wedge \text{swrlb} : \text{lessThan}(?a, 25) \rightarrow \text{UnderAge}(?x). \quad (7.8)$$

Here, citizenAge is a Data Property that indicates the age of a citizen in the network. The rule above depicts that citizens having age less than 25 belong to the class UnderAge.

Knot class contains the citizens grouped as specific names. This class, so, contains sub-classes such as Bank, Parents etc. These classes need to be defined using SWRL besides their OWL definitions. Before defining them, let us first take a look at the properties (Data and

Object) present in the ontology.

We have already mentioned a data property `citizenAge`, which is the only data property in our ontology. The object properties are:

`belongsTo`: is used to express a content's possession.

`hasViewer`: used for displaying citizens having view only permission to a content.

`hasRegulator`: gives a list of citizens having regulatory permission to a content.

`canRegulate`: gives a list of content a citizen can regulate.

`canView`: Being similar to `canRegulate`, `canView` gives a list of viewers instead of regulators.

`hasContentType`: tells whether a content is Sensitive or Common.

`hasRelationshipType`, `hasRelationWith`: show the relationship type a citizen holds and with whom.

`isfatherOf`, `isMotherOf`, `isParentOf`: related to each other so that any citizen having the `isFatherOf` or `isMotherOf` property is automatically included in the Parents knot and holds the `isParentOf` property. These properties show a citizen is Father or Mother of whom.

`isChildOf`: tells a citizen is child of whom.

`isSiblingOf`: When more than one citizen have the same Parents, they are connected with the `isSiblingOf` property showing who is sibling of whom.

`isFriendOf`, `isDoctorOf`, `isPatientOf`: These properties are associated with the other supported relationship types (Friend, Doctor, Patient, respectively) mentioned ear-lier.

`hasAccountWith`: associated with bank accounts of a citizen and tells which citizen has an account with which bank.

Now, let us get back to the classes of CBAC ontology. The Parents class under Knot uses the following SWRL definition:

$$\begin{aligned}
 & \text{Citizen}(?x) \wedge \text{Citizen}(?y) \wedge \text{hasRelationWith}(?x, ?y) \wedge \\
 & \quad ((\text{hasRelationshipType}(?x, \text{Father}) \wedge \\
 & \quad \quad \text{isFatherOf}(?x, ?y)) \vee \\
 & \quad (\text{hasRelationshipType}(?x, \text{Mother}) \wedge \\
 & \quad \quad \text{isMotherOf}(?x, ?y))) \rightarrow \\
 & \quad \text{Parents}(?x) \wedge \text{isParentOf}(?x, ?y) \wedge \\
 & \quad \text{hasRelationshipType}(?y, \text{Child}) \wedge \\
 & \quad \quad \text{isChildOf}(?y, ?x). \quad (7.9)
 \end{aligned}$$

If we look at the definition above, we see that if two citizens x and y are related to each other through the `hasRelationWith` property with a relation type `Father` or `Mother` and thus satisfy the `isFatherOf` or `isMotherOf` property, respectively, then y belongs to the `Parents` class. As a result, this also implies that y satisfies the `isParentOf` property of x and that x has a `Child` relation type with y .

Now comes the last subclass, `Associate`, of `Citizen`. It has two more subclasses, named `Regulator` and `Viewer`. Unless defined otherwise, the default `Regulator` rule for Sensitive content and default `Viewer` rule for Common content are, respectively:

$$\begin{aligned}
 & \text{Content}(?c) \wedge \text{belongsTo}(?c, ?x) \wedge \\
 & \quad \text{hasContentType}(?c, \text{Sensitive}) \rightarrow \\
 & \quad \quad \text{Regulator}(?x) \wedge \text{canRegulate}(?x, ?c) \wedge \\
 & \quad \quad \quad \text{hasRegulator}(?c, ?x). \quad (7.10)
 \end{aligned}$$

$$\begin{aligned}
& \text{Content}(?c) \wedge \text{Citizen}(?x) \wedge \\
& \quad \text{belongsTo}(?c, ?x) \wedge \\
& \quad \text{hasContentType}(?c, \text{Common}) \wedge \text{Citizen}(?y) \wedge \\
& \quad \text{hasRelationWith}(?x, ?y) \rightarrow \\
& \quad \text{Viewer}(?y) \wedge \text{hasViewer}(?c, ?y) \wedge \\
& \quad \text{canView}(?y, ?c). \quad (7.11)
\end{aligned}$$

The RelationType class has the supported relations in CBAC as instances. Any citizen having relation with other citizen(s) must select a relation type from this class. Next are the Content and ContentType classes. Individuals of Content class must have a type selected from the ContentType class. As mentioned before, the type must be one of the two members of the ContentType class: Common or Sensitive.

At this point, we are done with all the classes, subclasses and their relevant properties and rules defined in the CBAC Ontology. Also, we have specified the necessary policies for access control in the NID Network. We can now run some experiments to see if our ontology works correctly satisfying the policies. For this, we need to look into the next chapter describing some experimental cases.

Chapter 8

Experiment

In this chapter, we present some experimental cases we have performed with our CBAC model using the ontology in order to substantiate the validity of our proposed methodology. The experimental setup, including different variations (cases) and corresponding results are described below.

8.1 Experimental Setup

The experiment was run using Protégé [65, 66], which is an open-source ontology editor as well as framework for intelligent system building. Specifically, Protégé is a multi-platform java-based application which has a GUI for a series of ontology editing (creation, modification, reasoning, debugging etc.) which made us choose Protégé for our experimental works. We used Protégé version 5.2.0 for our work.

We have declared the required classes and subclasses mentioned in Section 7.2 using the editor. All the classes can be seen under the ‘class hierarchy’ tab as seen in Figure 8.1.

The graphical representation of all the classes can be obtained under the ‘OntoGraf’ window which gives a clearer view of the classes and subclasses (Figure 8.2). We can expand any class to instances here, but for the ease of viewing, we have expanded only to subclasses.

Figure 8.3 shows all the object properties mentioned in section 7.2. Figure 8.4(a) shows the supported relationships in CBAC ontology while Figure 8.4(b) shows the content types. We have declared these as instances of the classes RelationType and ContentType, respectively.



Figure 8.1: Class Hierarchy in Protégé.

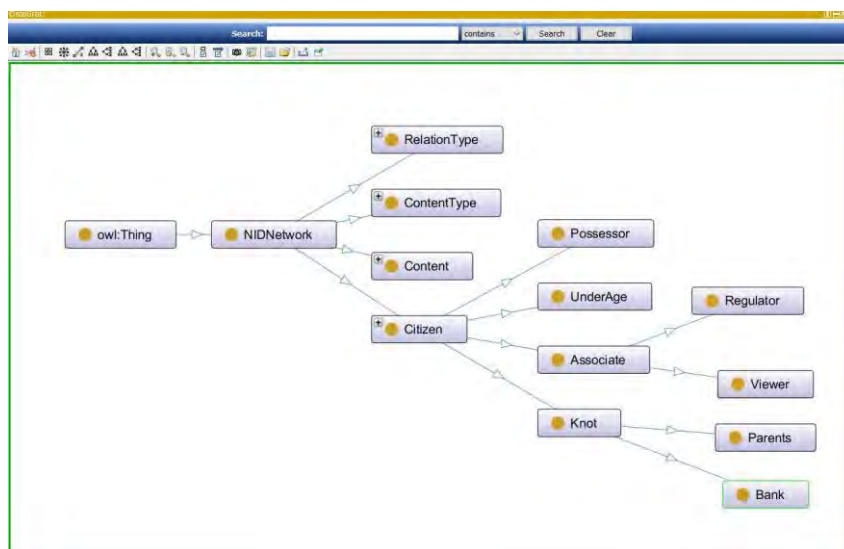


Figure 8.2: Graphical Representation of the Class Hierarchy in Protégé. Only up to to sub-classes view has been presented.

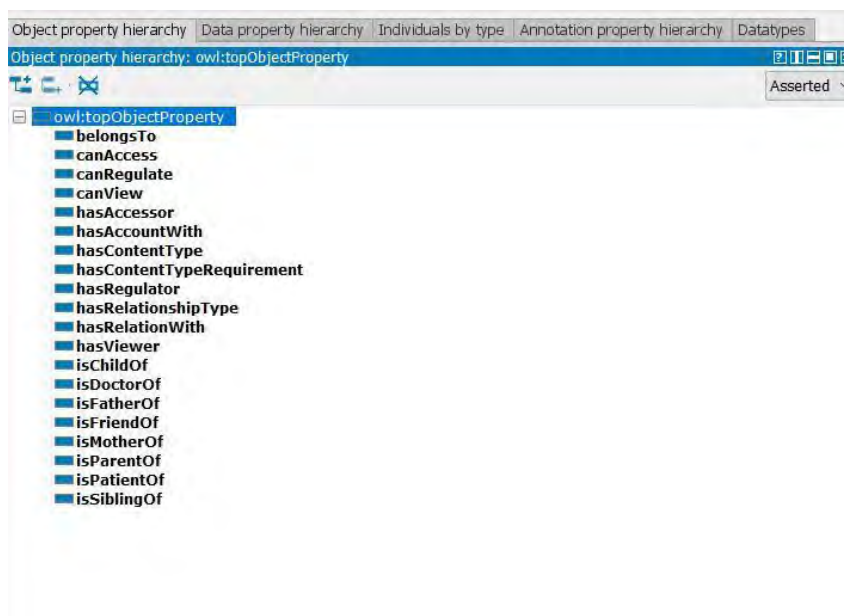


Figure 8.3: Object Properties of the CBAC Ontology in Protégé.

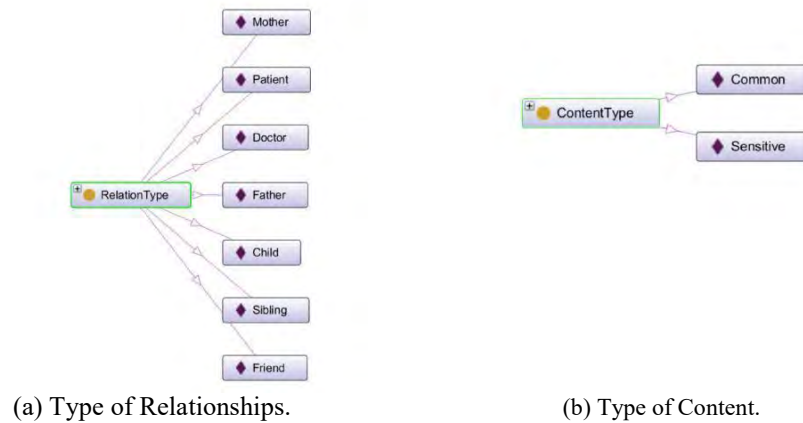


Figure 8.4: Ontograf Showing Relationship Types and Content Types of CBAC Ontology.

8.2 Cases

In our experiment, we have picked up three pertinent cases for testing the CBAC rules and policies for content sharing. We have defined different content types, each with different sharing demand, in these cases and tested whether the rules defined within the ontology works correctly. In the very first case, we have considered the default setting of a content which has a content type Common.

The next case describes a bit restricted settings for another content that is considered as sensitive to the user. Finally, the last case deals with more complex settings including two parties. We have chosen these three cases because every other scenario can be described using these three or their modified versions. We have also defined a number of individuals in Protégé for assigning different contents. Some contents have been declared under the Content class (Figure 8.5(a)) and the ones possessing/regulating/viewing them are the individuals under the Citizen class (Figure 8.5(b)).

Having known all the required individuals and contents, the experimental cases are now described below:

8.2.1 Common Content

Let us suppose that a citizen B has two more citizens X and T in her loop. She sets her NID Number ($B.NIDNo$) as Common. According to the default rule mentioned in Section 7.2, the content should be visible to all the citizens in her loop. Figure 8.6(a) along with the SPARQL query result in Figure 8.6(b) confirm that $B.NIDNo$ has T and X as its viewers.

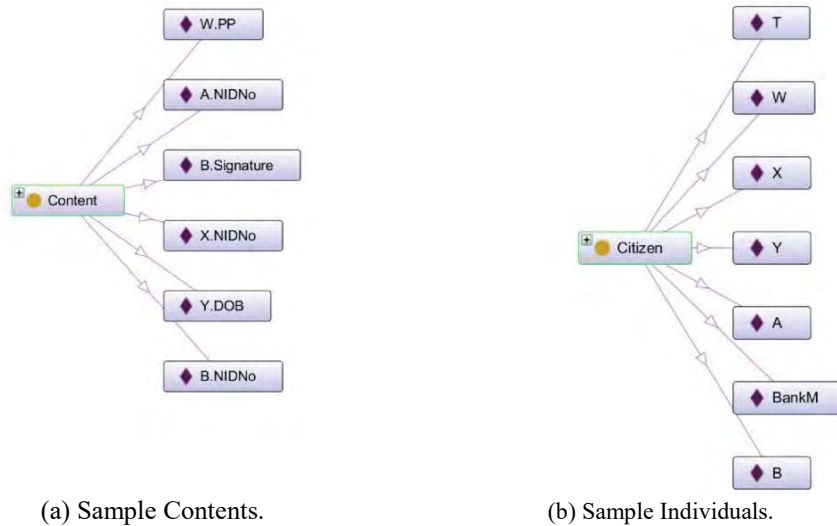
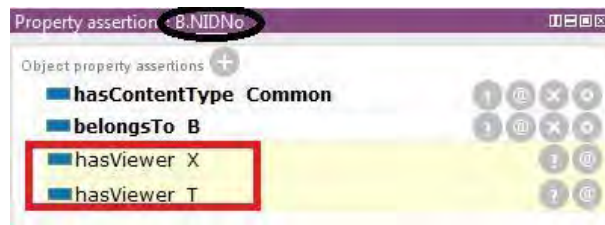
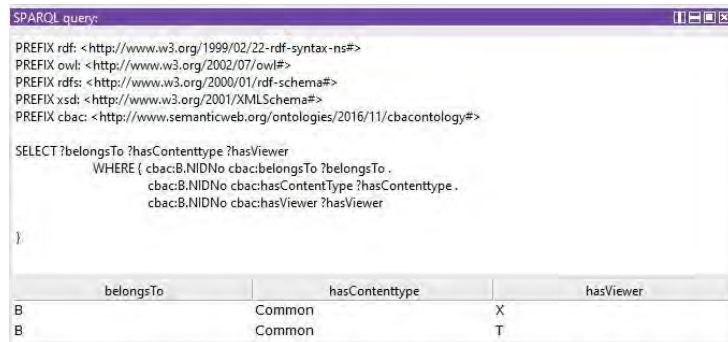


Figure 8.5: Sample Contents and Individuals for Experimental Cases of CBAC Ontology.



(a) Property assertion window of B.NIDNo.



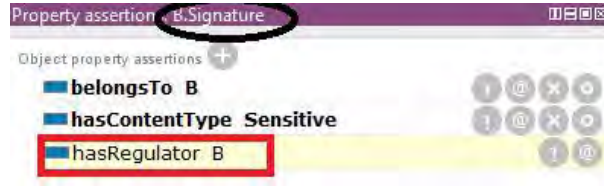
(b) SPARQL query result for B.NIDNo.

Figure 8.6: Experimental result validating default sharing for common content.

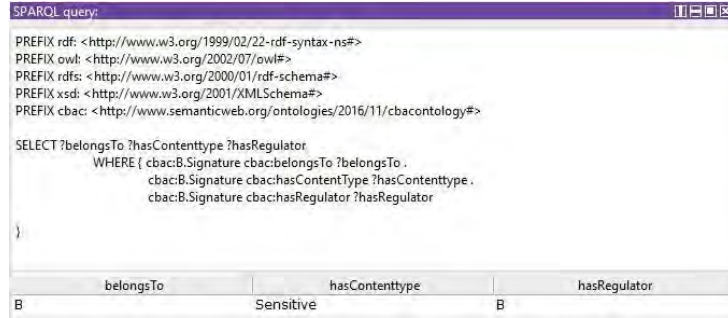
8.2.2 Sensitive Content

Now, *B* marks her signature (*B.Signature*) as Sensitive. So, according to the default rule, the content is visible to no one in her loop and has only one regulator, i.e., the possessor, *B*. As can be seen in Figure 8.7(a), *B.Signature* shows only *B* as the value of the *hasRegulator* property.

The SPARQL query selecting the required properties of *B.Signature* also gives the same



(a) Property assertion window of B.Signature.



(b) SPARQL query result for B.Signature.

Figure 8.7: Experimental result validating default sharing for sensitive content.

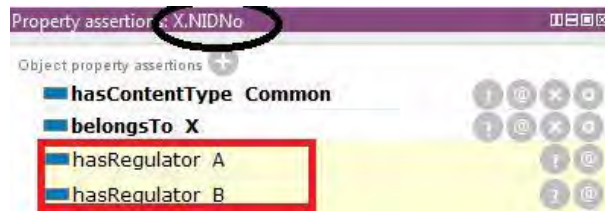
result as shown in Figure 8.7(b). Thus, it satisfies the policy for Sensitive content mentioned in Equation (7.1).

8.2.3 Supervisory Content Settings

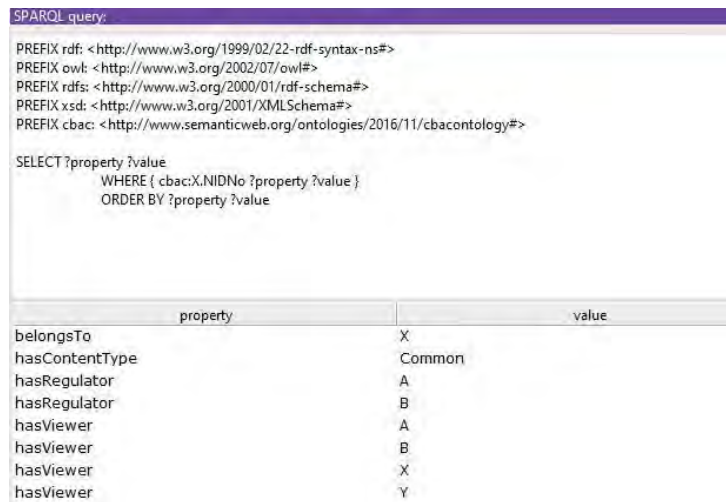
Now, National ID is provided from the age of 15 which is still mid-teenage. When it comes to content sharing, NID Network needs more security than traditional OSNs because all information present in this network can be subject to forgery. So parents may want to control their children's content sharing at least to some extent. CBAC model supports this feature by introducing age limit to the system hence content sharing rule. This model gives parents Regulate permission to their children's contents to a certain age limit so that the best privacy settings common to both parties (parents and children) are applied. In the CBAC ontology, the automated access of parents to the contents of an underage citizen has been achieved with the following rule:

$$\begin{aligned}
 & \text{Citizen}(?x) \wedge \text{UnderAge}(?x) \wedge \text{Content}(?c) \wedge \text{belongsTo}(?c, ?x) \wedge \text{Parents}(?y) \wedge \\
 & \text{isParentOf}(?y, ?x) \rightarrow \text{Regulator}(?y) \wedge \text{hasRegulator}(?c, ?y) \wedge \\
 & \text{canRegulate}(?y, ?c). \quad (8.1)
 \end{aligned}$$

Let us make it clearer with another experimental case. Let us assume that X is 23 years of



(a) Property assertion window of X.NIDNo.



(b) SPARQL query result for X.NIDNo.

Figure 8.8: Experimental result validating special content settings.

age and has *A* and *B* as father and mother, respectively. CBAC has an adult age limit set for 25 years. So, *X* belongs to the UnderAge class. She has her NID Number (*X.NIDNo*) published in the network.

Figure 8.8(a) shows that shows that *A* and *B* are automatically inferred as regulators of *X.NIDNo* hence satisfies the policy stated in Equation (7.5). Figure 8.8(b) shows the relevant SPARQL query that gives the same output.

Having described the experimental cases, we can see that they satisfy the policies specified in Section 7.1. Thus we finish implementing the content sharing policies for our proposed model.

Chapter 9

Focused Challenges

Gaining detail knowledge about the our proposed CBAC model and experiencing it through various cases, let us now discuss the issues it resolves through its privacy mechanism.

9.1 Introducing New Attribute: Content Type

As mentioned in Section 2.4, controlling access in online communities based on content type is one of the major challenges. CBAC model works mainly on content types. There are a number of works that take into account the content type and present tag-based access control policies [67–69]. According to these works, privacy policies are expressed in terms of tags on content objects by the content owners. The system then applies the policy to objects based on the tags assigned to them.

Unlike tag-based policy, we have proposed class-based policy for access control over content sharing. While introducing content type as an attribute, increase in complexity is envisaged as a major issue in some publications [29]. A user may define n number of tags for her shared contents which leads to a complexity of $O(n)$. On the other hand, according to our proposal, a content can only be of two types: Sensitive or Common. So, the complexity of content sharing in CBAC reduces to $O(2)$.

Also, if we compare the CBAC model with the MPAC model presented in [4], we see that introducing the new attribute doesn't increase number of policies. Rather users are able to control their contents only classifying them as Sensitive or Common.

9.2 Other Focused Challenges

Although our proposed model takes mainly into account the content type in any online community, it has focused on some other challenges (mentioned in Section 2.4) too which will make the NID Network as well as OSNs more efficient, useful and privacy safe.

9.2.1 Supervisory Settings

Figure 8.8(a) in Chapter 8 shows an example of supervisory setting for the content of citizen X . This can be also applied to traditional OSNs because personal information and photo shared by users can also be misused by third parties and certainly parents would want to monitor their children's safety while sharing anything in social networks.

9.2.2 Sticky Policies and Co-privacy

Sticky policies are those that stick to the shared information even if it moves or is moved to different contexts. These policies enable users to improve control over their personal information as it moves across multiple parties [29]. In CBAC, when a person is tagged or mentioned in a content and marked as Sensitive by the owner, the content shows itself to the tagged user's space but not to any of the people in her circle that is not mutual between the owner and the tagged user.

Let us suppose that P mentions Q in one of her contents c_1 . She marks c_1 as Sensitive and chooses G , H and K from her loop to view it. Now, in Q 's space, c_1 is visible to only Q . If anyone or all users from P 's sharing policy is in Q 's circle, they can view it. But no one else from Q 's list can view c_1 . Again, suppose P marks another content c_2 as Sensitive and chooses only Q to view it. Now, Q wants to share c_2 in her own space. But even if she does so, the policy P originally applied to it, shall still apply. As a result, Q 's loop can not view c_2 . CBAC will thus focus on policies to stick with a bit more strictness as for the safety of a user's sensitive identifying information.

While handling sticky policies according to owner's preference, there comes a question of co-privacy of the tagged users in a content. When a content is marked as Common with two other users mentioned with it, they can naturally want the content to be shared with only their chosen people. According to CBAC, tagged users can request Regulate permission to the particular

content and if the owner grants it, they can choose the settings of how to share it on their own spaces. Generally the mentioned users won't be given the privilege to control the content in owner's space, as it will be a breach to owner's privacy. But in CBAC, a Decision-voting mechanism is followed to control the sensitivity of a shared content with tagged users in it. Let us suppose, P again shares a content c_3 mentioning Q , R , S and Z and marks it as Common. By de-fault, the content will be shared with everyone in P 's loop and of course, in Q , R , S , Z 's spaces with their own settings of sharing contents. But the information is sensitive to R , S and Z and they want c_3 to be shared with none other than themselves. So they request Regulate permission over c_3 and let us assume that P grants the requests. Now, R , S and Z marks c_3 to be Sensitive. CBAC, using Boyer-Moore Algorithm [70] checks whether the number of voting in support of Sensitive is more than half of the participants, i.e., stakeholders, of c_3 . The system keeps an array for each tagged content and takes input the decisions from all the participants of the content. So, for content c_3 the inputs will be, {Common, Common, Sensitive, Sensitive, Sensitive}. We see that among the five votes, three, i.e., more than half of the votes are Sensitive. So, following the algorithm in [70], c_3 is automatically marked as Sensitive for both owner and stakeholders and is shared with none but themselves. It is to be remembered that the Regulatory control for making a content Sensitive in all the owner and stakeholder(s)'s space is applicable when the tagged content is marked as Common by the owner and it is different from the Supervisory Access described in Section 9.2.1.

9.2.3 Knowledge Reusing

As described in Section 7.2, we have developed an ontology for the CBAC model. The advantage of this ontology, like all other ontologies, is that it allows to share common understanding and reuse the domain knowledge. This domain knowledge is generic to all type of online communities. This ontology can be reused from time to time for various kinds of online communities besides the NID Network which gives the CBAC model a general view. As a result, interoperability of privacy settings can be achieved through the reuse of domain knowledge.

Since this model doesn't focus on user relationships, so we didn't address the challenges of relationship model and strength in this research. Apart from those, the issues of sticky policies, co-privacy and reusable privacy settings are successfully dealt by this model along with the major challenge of handling the type of the content and setting up rules according to the

content type. Summarizing these, the proposed model can be considered a competitive access control model resolving a number of open areas in the field of privacy mechanisms in online communities.

Chapter 10

Conclusion and Future Work

10.1 Compendium of Attainments

In our research, we have proposed a new access control system, CBAC, which is suitable for online communities and takes into account the sensitivity of content shared by the users. A brief analytic view of the proposed model is given below.

- The proposed model introduces a new attribute, CnT, which represents the type of a content depending on its sensitivity. Because, regardless of the purpose of online communities, the major reason for people to participate is to share and exchange content or information with other users. So the security of this content is important. Especially, in a national database, sharing of these information might affect the security of an user. So we focused on the type of the uploaded data and proposed the CBAC model with a view to keeping the contents of the network users safe and secured.
- The proposed system won't be too much complex for NID Network as the number of rules don't increase. Because, there is only one rule for content-sharing which can be updated according to the purpose of the community, but not necessarily needs to be increased.
- Our system takes up a semantic approach for which we have created an ontology with classes, individuals, properties and relations. We have also defined all access control rules in the ontology using SWRL.
- We have experimented our proposed model with three different cases and satisfied all the specified privacy policies both in asserted property outputs and SPARQL queries.

- The proposed model is able to deal with sticky policies and co-privacy along with the supervisory content settings that helps users setting up a common policy where more than one parties are interested in controlling the privacy of a shared content. The sticky policies and co-privacy of the model covers a technical challenge that needs to be meet for sharing content safely. Again, a supervisory setting adds an extra layer of safety on content sharing by underage citizen.
- Lastly, the reusable ontology helps providing a privacy setting which is interoperable among online communities.

10.2 Future Work

In future work of the CBAC model, we would like to focus on the security of the system from the threats existing in the world of Web. Because the NID Network is a significant and sensitive online community holding all its users' genuine information, our future work includes ensuring secrecy of the semantic model using RDF and XML layers for data encryption and enhanced authentication of shared content. Finally, the future work focuses on enriching the ontology and developing a semantic database for the NID Network that will serve even better with the ontology.

Bibliography

- [1] Y. Cheng, J. Park, and R. Sandhu, “A user-to-user relationship-based access control model for online social networks,” in *Data and Applications Security and Privacy XXVI: 26th Annual IFIP WG 11.3 Conference, DBSec 2012, Paris, France, July 11-13,2012. Pro-ceedings* (N. Cuppens-Boulahia, F. Cuppens, and J. Garcia-Alfaro, eds.), pp. 8–24, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [2] Y. Cheng, J. Park, and R. Sandhu, “Relationship-based access control for online social networks: Beyond user-to-user relationships,” in *2012 International Conference on Pri-vacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pp. 646–655, Sept. 2012.
- [3] B. Carminati, E. Ferrari, and A. Perego, “Rule-based access control for social networks,” in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pp. 1734– 1744, Springer, 2006.
- [4] H. Hu, G.-J. Ahn, and J. Jorgensen, “Multiparty access control for online social net-works: Model and mechanisms,” *IEEE Transactions on Knowledge and Data Engineer-ing*, vol. 25, pp. 1614–1627, July 2013.
- [5] E. Miller, “An introduction to the resource description framework.” <http://www.dlib.org/dlib/may98/miller/05miller.html>, May 1998. [Online; Ac-cessed 12 March, 2018].
- [6] L. Yu, *A developer’s guide to the semantic Web*. Springer-Verlag Berlin Heidelberg, 2014.
- [7] P. Hitzler, M. Krotzsch, and S. Rudolph, *Foundations of semantic web technologies*. CRC Press, 2009.
- [8] “OWL 2 web ontology language.” <https://www.w3.org/TR/owl2-overview/>.

- [9] M. O'Connor, "The semantic web rule language," in Protégé conference, 2009.
- [10] H. Alhazmi, S. S. Gokhale, and D. Doran, "Understanding social effects in online net-works," in Computing, Networking and Communications (ICNC), 2015 International Conference on, pp. 863–868, Feb 2015.
- [11] H. Alhazmi and S. S. Gokhale, "Analysis of structural social capital in online social net-works," in Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on, pp. 473–480, IEEE, 2015.
- [12] R. S. Sandhu and P. Samarati, "Access control: principle and practice," IEEE communications magazine, vol. 32, no. 9, pp. 40–48, 1994.
- [13] P. W. L. Fong and I. Siahaan, "Relationship-based access control policies and their policy languages," in Proceedings of the 16th ACM symposium on Access control models and technologies(SACMAT '11), (Innsbruck, Austria), pp. 51–60, ACM, 2011.
- [14] B. Carminati and E. Ferrari, "Collaborative access control in on-line social networks," in Collaborative computing: networking, applications and worksharing (CollaborateCom), 2011 7th international conference on, pp. 231–240, IEEE, 2011.
- [15] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214–1221, 2011.
- [16] A. Ahmad and B. Whitworth, "Distributed access control for social networks," in Information Assurance and Security (IAS), 2011 7th International Conference on, pp. 68–73, IEEE, 2011.
- [17] T. Abdessalem and I. B. Dhia, "A reachability-based access control model for online social networks," in Databases and Social Networks, pp. 31–36, ACM, 2011.
- [18] A. Masoumzadeh and J. Joshi, "Ontology-based access control for social network systems," International Journal of Information Privacy, Security and Integrity, vol. 1, no. 1, pp. 59–78, 2011.

- [19] X. Jin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering dac, mac and rbac," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 41–55, Springer, 2012.
- [20] E. E. Mon and T. T. Naing, "The privacy-aware access control system using attribute-and role-based access control in private cloud," in *Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE International Conference on*, pp. 447–451, IEEE, 2011.
- [21] J. Park, R. Sandhu, and Y. Cheng, "A user-activity-centric framework for access control in online social networks," *IEEE Internet Computing*, vol. 15, no. 5, pp. 62–65, 2011.
- [22] M. Vanetti, E. Binaghi, B. Carminati, M. Carullo, and E. Ferrari, "Content-based filtering in online social networks," in *Privacy and Security Issues in Data Mining and Machine Learning: International ECML/PKDD Workshop, PSDML 2010, Barcelona, Spain, September 24, 2010. Revised Selected Papers* (C. Dimitrakakis, A. Gkoulalas-Divanis, A. Mitrokotsa, V. S. Verykios, and Y. Saygin, eds.), pp. 127–140, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [23] B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, p. 6, 2009.
- [24] S. H. P. Oo, "Intelligent access control policies for social network site," *International Journal of Computer Science & Information Technology*, vol. 5, no. 3, p. 183, 2013.
- [25] S. B. Barnes, "A privacy paradox: Social networking in the united states," *First Monday*, vol. 11, no. 9, 2006.
- [26] A. L. Vangelisti and D. Perlman, *The Cambridge handbook of personal relationships*. Cambridge University Press, 2006.
- [27] T. Berners-Lee, D. Connolly, S. Palmer, and M. Nottingham, "Cwm—a general purpose data processor for the semantic web, 2000," URL: <http://www.w3.org/2000/10/swap/doc/cwm.html>, vol. 16, 2012.

- [28] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler, “N3logic: A logical framework for the world wide web,” *Theory and Practice of Logic Programming*, vol. 8, no. 3, pp. 249–269, 2008.
- [29] R. Fogues, J. M. Such, A. Espinosa, and A. Garcia-Fornes, “Open challenges in relationship-based privacy mechanisms for social network services,” *International Journal of Human-Computer Interaction*, vol. 31, no. 5, pp. 350–370, 2015.
- [30] J. M. Such, A. Espinosa, A. García-Fornes, and C. Sierra, “Self-disclosure decision making based on intimacy and privacy,” *Information Sciences*, vol. 211, pp. 93–111, 2012.
- [31] “W3C semantic web activity.”
- [32] G. Antoniou, P. Groth, F. van Harmelen, R. Hoekstra, et al., “A semantic web primer.-,” in *Cooperative information systems*, MIT Press, 2012.
- [33] P. Hitzler, M. Krtzsch, and S. Rudolph, *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 1st ed., 2009.
- [34] T. Berners-Lee, J. Hendler, O. Lassila, et al., “The semantic web,” *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.
- [35] “Linked data.” <https://www.w3.org/DesignIssues/LinkedData.html>, July 2006. [Online; accessed 06-March-2018].
- [36] “Resource description framework (rdf).”
- [37] T. Berners-Lee, L. Masinter, and M. McCahill, “Uniform resource locators (url).” RFC 1738, 1994.
- [38] T. Bray, D. Hollander, and A. Layman, “Namespaces in xml 1.0.” <https://www.w3.org/TR/xml-names/>, Dec 2009.
- [39] “Dublin core metadata initiative.” <http://dublincore.org/>.
- [40] “Description of the dublin core elements.” <http://dublincore.org/documents/1998/09/dces/>, Sep 1998.
- [41] H. Halpin, R. Iannella, B. Suda, and N. Walsh, “Representing vcard objects in rdf.” <https://www.w3.org/Submission/vcard-rdf/>, Dec 2010.

- [42] F. Gandon and G. Schreiber, “Rdf 1.1 xml syntax.” <https://www.w3.org/TR/rdf-syntax-grammar/>.
- [43] D. Beckett, T. Berners-Lee, E. Prud’hommeaux, G. Carothers, and L. Machina, “Terse rdf triple language.” <https://www.w3.org/TR/turtle/>.
- [44] M. Birbeck, S. Pemberton, and B. Adida, “Rdfa syntax.” <https://www.w3.org/2006/07/SWD/RDFa/syntax/>.
- [45] “OWL web ontology language use cases and requirements.” <https://www.w3.org/TR/webontreq/>, 2004. [Online; accessed 01-March-2018].
- [46] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, “An environment for merging and testing large ontologies,” in KR, pp. 483–493, 2000.
- [47] J. Heflin, “OWL web ontology language-use cases and requirements,” W3C Recommendation, vol. 10, p. 12, 2004.
- [48] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, et al., “Owl 2 web ontology language profiles,” W3C recommendation, vol. 27, p. 61, 2009.
- [49] J. M. Crawford and B. Kuipers, “Negation and proof by contradiction in access-limited logic.,” in AAAI, pp. 897–903, 1991.
- [50] T. Berners-Lee, “The semantic web as a language of logic.” <https://www.w3.org/DesignIssues/Logic.html>, 2009.
- [51] M. H. Van Emden and R. A. Kowalski, “The semantics of predicate logic as a programming language,” Journal of the ACM (JACM), vol. 23, no. 4, pp. 733–742, 1976.
- [52] R. Kowalski, “Predicate logic as programming language,” in IFIP congress, vol. 74, pp. 569–544, 1974.
- [53] “Horn clause logic.” <https://cs.nyu.edu/courses/spring02/G22.2560-001/horn.html>.
- [54] G. Antoniou and F. Van Harmelen, A semantic web primer. MIT press, 2004.
- [55] Wikipedia, “Bangladeshi national identity card — wikipedia, the free encyclopedia,” 2017. [Online; accessed 29-August-2017].

- [56] N. E. News, “Nid bd smart card election commission bd,” 2017. [Online; accessed 29-August-2017].
- [57] “Id card frequently asked questions.” <https://web.archive.org/web/20110903074029/https://www.privacyinternational.org/article/id-card-frequently-asked-questions>, August 1996.
- [58] Wikipedia contributors, “List of national identity card policies by country — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=List_of_national_identity_card_policies_by_country&oldid=836046404, 2018. [Online; accessed 17-April-2018].
- [59] R. B. Black, “Legislating US data privacy in the context of national identification numbers: models from south africa and the united kingdom,” *Cornell Int’l LJ*, vol. 34, p. 397, 2001.
- [60] Jon Russell, “India’s national ID database is reportedly accessible for less than usd10.” <https://techcrunch.com/2018/01/04/indias-national-id-database-is-reportedly-accessible-for-less-than-Jan>. 2018. [Online; accessed 16-April-2018].
- [61] A. Shontell, “How to figure out exactly what day — and in what order — you signed up for facebook,” 2014. [Online; accessed 31-August-2017].
- [62] InfoZone24, “What is the meaning of 13 digits of bd nid?,” 2016. [Online; accessed 31-August-2017].
- [63] “Election commission — Bangladesh.” https://services.nidw.gov.bd/voter_center, 2014.
- [64] D. Brickley and L. Miller, “Foaf vocabulary specification 0.99,” 2014.
- [65] M. Musen, “The Protégé project: A look back and a look forward,” *AI Matters*, vol. 1, pp. 4–12, June 2015.
- [66] <http://protege.stanford.edu/>.
- [67] P. Klemperer, Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. F. Cranor, N. Gupta, and M. Reiter, “Tag, you can see it!: using tags for access control in photo sharing,” in Pro-

ceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 377–386, ACM, 2012.

- [68] C.-m. A. Yeung, L. Kagal, N. Gibbins, and N. Shadbolt, “Providing access control to online photo albums based on tags and linked data.,” in *AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0*, pp. 9–14, 2009.
- [69] M. Hart, C. Castille, R. Johnson, and A. Stent, “Usable privacy controls for blogs,” in *Computational Science and Engineering, 2009. CSE’09. International Conference on*, vol. 4, pp. 401–408, IEEE, 2009.
- [70] R. S. Boyer and J. S. Moore, “A fast string searching algorithm,” *Communications of the ACM*, vol. 20, no. 10, pp. 762–772, 1977.