A Privacy Preference Manager for the Social Semantic Web*

Owen Sacco and Alexandre Passant

Digital Enterprise Research Institute,
National University of Ireland, Galway, Ireland
{firstname.lastname}@deri.org
http://www.deri.ie/

Abstract. Current Social Web applications provide users with means to easily publish their personal information on the Web. However, once published, users cannot control how their data can be accessed apart from applying generic preferences (such as "friends" or "family"). In this paper, we describe how we enable finer-grained privacy preferences using the Privacy Preference Ontology (PPO); a light-weight vocabulary for defining privacy settings on the Social Web. In particular, we describe the formal semantic model of PPO and also present *MyPrivacyManager*, a privacy preference manager that let users (1) create privacy preferences using the aforementioned ontology and (2) restrict access to their data to third-party users based on profile features such as interests, relationships and common attributes.

1 Introduction

In the past few years, the growing number of personal information shared on the Web (through Web 2.0 applications) increased awareness regarding privacy and personal data. A recent study [2] showed that privacy in Social Networks is a major concern when private news are publicly shared, revealing that most users are aware of privacy settings and have set them at least once since 2009.

Most Social Networks provide privacy settings restricting access to private data to those who are in the user's friends lists (i.e. their "social graph") such as Facebook's privacy preferences and Google+ circles. Yet, the study shows that users require more complex privacy settings as current systems do not meet their requirements.

We aim to solve these privacy shortfalls with the Privacy Preference Ontology (PPO)[1]. This model can be applied to any social data as long as it is modelled in RDF (for instance using FOAF[2], SIOC[3] or OGP[4] can be used to

* This work is funded by the Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2) and by an IRCSET scholarship co-funded by Cisco systems.

[1] http://vocab.deri.ie/ppo#

[2] Friend-of-a-Friend — http://www.foaf-project.org

[3] Semantically-Interlinked Online Communities - http://sioc-project.org/

[4] Open Graph Protocol - http://ogp.me/

define such fine-grained settings. While data from major websites is generally not modelled directly in RDF, wrappers can easily be implemented through their API. In addition, PPO can be natively used in Social Semantic Web applications, i.e. Social Web applications directly using RDF to model their data, such as Semantic MediaWiki or Drupal 7.

In this paper, we detail the formal model of PPO, and also present a privacy preference manager (*MyPrivacyManager*), letting users: (1) create privacy preferences described using PPO for their FOAF profiles; and (2) view other user's profiles, filtered according to their privacy preferences.

The remainder of the paper is organised as follows: Section 2 provides an overview of the Privacy Preference Ontology (PPO) and presents use cases for PPO. In Section 3, we present our formal model. In section 4 we present the implementation of *MyPrivacyManager*. Section 5 discusses related work and Section 6 presents future work and concludes the paper.

2 The Privacy Preference Ontology (PPO)

2.1 Overview

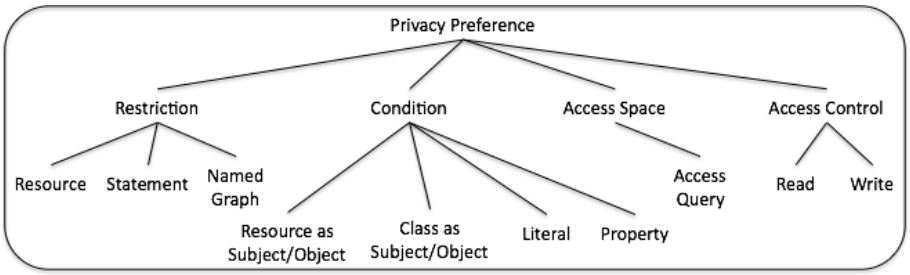

Fig. 1. A high level graphical representation of the properties that make up a privacy preference.

The Privacy Preference Ontology (PPO) [10] provides a light-weight vocabulary enabling Linked Data creators to describe fine-grained privacy preferences for restricting (or granting) access to specific data. PPO can be used for instance to restrict part of a FOAF user profile only to users that have similar interests. It provides a machine-readable way to define settings such as "Provide my phone number only to DERI colleagues" or "Grant write access to this picture gallery only to people I've met in real-life".

As we deal with Semantic Web data, a privacy preference (Figure 1), defines: (1) which resource, statement or named graph to restrict access to ; (2) the conditions to refine what to restrict; (3) the access control type; and (4) a SPARQL query, known as an `AccessSpace` containing a graph pattern representing what

must be satisfied by the user requesting information. The access control type is defined by using the Web Access Control (WAC)[5] vocabulary which defines the `Read` and `Write` access control privileges (for reading or updating data).

2.2 Use Case

As mentioned in section 1, current social networks provide minimum privacy settings such as granting privileges to all people belonging to one's social graph to access his/her information. Suppose a social network which provides users to specify which information can be accessed by specific users not necessarily in one's social graph, for instance having similar interests. Although applications are being developed to export user information from closed social networks into RDF, the privacy settings are platform dependent such that the privacy settings cannot be reused on other platforms. Moreover, privacy preferences cannot make use of other platform's information, for instance, defining a privacy preference that restricts access to users from one platform and grants users from another platform. Therefore, a system is required that provides users to create fine-grained privacy preferences described using PPO which can be used by different platforms. This system will provide users to be fully in control who can access their personal information and who can access their published RDF data. Additionally, the user can set privacy preferences to control which data can be used by recommender systems or other applications.

3 A Formal Model for the Privacy Preference Ontology (PPO)

As portrayed in figure 1, a PPO-based privacy preference consists of: (1) `Restrictions`; (2) `Conditions`; (3) `Access Control Privileges` and; (4) `Access Spaces`. This section presents the associated formal model for PPO.

3.1 Defining the Classes and Properties of PPO

Definition 1: Restrictions. A `restriction` applies to a *Resource*, a *Statement* or a *Named Graph* (Fig. 1), where:

- A *Resource* (instance of `rdfs:Resource`) is identified by its own URI;
- A *Statement* consists of a $< subject, predicate, object >$ triple, each being instances of `rdfs:Resource`[6];
- A *Named Graph* consists of (1) a name denoted by a URI, and (2) a set of statements (an RDF graph) mapped to this name [4].

[5] WAC — `http://www.w3.org/ns/auth/acl`
[6] Including literals

Let St be a statement, U a URI, S be a subject, P a predicate, O an object, NG a named graph and A an access control privilege. Let $Subject(U, St)$ mean that U is subject of St, $Predicate(U, St)$ mean that U is a predicate of St, $Object(U, St)$ mean that U is an object of St, $RDFGraph(St, NG)$ mean that St is contained within the RDF graph of NG and $AssignAccess(U, A)$ mean that A is assigned to U.

Restricting access to a resource is defined as follows.

$$\forall St(AssignAccess(U,A) \wedge (Subject(U,St) \vee Predicate(U,St) \vee Object(U,St)) \Rightarrow AssignAccess(St,A)) \quad (1)$$

In other words, restricting access to a resource restricts access to all statements involving that resource as subject, predicate or object.

Restricting access to a statement is defined as follows.

$$\forall St((AssignAccess(S,A) \wedge AssignAccess(P,A) \wedge AssignAccess(O,A)) \wedge (Subject(S,St) \wedge Predicate(P,St) \wedge Object(O,St)) \Rightarrow AssignAccess(St,A)) \quad (2)$$

Restricting access to a named graph is defined as follows.

$$\forall St(AssignAccess(NG,A) \wedge RDFGraph(St,NG) \Rightarrow AssignAccess(St,A)) \quad (3)$$

In other words, restricting access to a Named Graph restricts access to all statements within that Graph.

Definition 2: Conditions. A `condition` defines whether what is being restricted has:

- a resource's URI identified as a statement's subject or object;
- an instance of a class which is defined as a statement's subject or object;
- a statement contains a particular literal as a value and;
- a statement that contains a particular property.

Let St be a statement, U a URI, C a class and A an access control privilege. Let $Subject(U, St)$ mean that U is subject of St, $Object(U, St)$ mean that U is the object of St, $RDFType(U, C)$ mean that U rdf:type C and AssignAccess(U,A) mean that A is assigned to U.

The condition resource as subject is defined as follows.

$$\forall St(AssignAccess(U,A) \wedge Subject(U,St) \Rightarrow AssignAccess(St,A)) \quad (4)$$

The condition resource as object is defined as follows.

$$\forall St(AssignAccess(U,A) \wedge Object(U,St) \Rightarrow AssignAccess(St,A)) \quad (5)$$

The condition class as subject is defined as follows.

$$\forall St(AssignAccess(C,A) \wedge RDFType(U,C) \wedge Subject(U,St) \Rightarrow AssignAccess(St,A)) \quad (6)$$

The condition class as object is defined as follows.

$$\forall St(AssignAccess(C,A) \wedge RDFType(U,C) \wedge Object(U,St) \Rightarrow AssignAccess(St,A)) \quad (7)$$

Definition 3: Access Control Privilege. An `access control privilege` defines the `read` and/or `write` privilege (defined by the WAC), and it is defined as:

$$AccessControl = \{read,write\}. \quad (8)$$

Definition 4: Access Space. An `Access Space` contains an access query that is executed to check whether a requester satisfies specific attributes. An access space can have multiple queries and therefore, it can be defined as the set:

$$AccessSpace = \{accessquery_1,...,accessquery_n\}. \quad (9)$$

3.2 Defining a Privacy Preference

Definition 5: A Privacy Preference. A privacy preference is the set of all the sets `Restrictions`, `Conditions`, `Access Control Privilege` and `Access Space` and it is defined as:

$$PrivacyPreference \subseteq Restrictions \cup Conditions \cup AccessControl \cup AccessSpace. \quad (10)$$

3.3 Applying Privacy Preferences

A privacy preference applies when requested information matches with the restricted statement(s), resource(s) and/or named graph(s). This is defined as follows. Let St be a requested statement, R a requested resource, NG a requested named graph and P a privacy preference. Let $ApplyPrivacyPreference(P)$ mean that P is applied, $Statement(St, P)$ mean that St is a restricted statement in P, $Resource(R, P)$ mean that R is a restricted resource in P and $NamedGraph(NG, P)$ mean that NG is a restricted named graph in P. Then:

$$\forall P((Statement(St,P) \vee Resource(R,P) \vee NamedGraph(NG,P)) \Rightarrow ApplyPrivacyPreference(P)) \quad (11)$$

The relationship between restrictions and conditions consists of a mapping from restricted statements RS to condition statements CS, which this mapping is defined as $M : RestrictedStatements(RS) \mapsto ConditionStatements(CS)$. `IF` M = `false THEN` ¬ApplyPrivacyPreference(P).

However, there are situations where restrictions are not defined but only conditions are defined within a privacy preference. In this case, the mapping

is performed between the RequestedInformation(RI) and the ConditionStatements(CS). This mapping is defined as *M : RequestedInformation(RI)* $\mapsto$ *ConditionStatements(CS)*. `IF` M = `true THEN` ApplyPrivacyPrefence(P). Therefore, applying a privacy preference based on the mapping between restricted or requested statements and condition statements is defined as: $\forall$*PM(P)* $\rightarrow$ *ApplyPrivacyPreference(P)*.

The access space query Q is executed on the requester's authenticated information. `IF` AccessSpace(Q) = `true THEN` AccessControl(A) defined in the privacy preference is granted to the requester. `IF` AccessSpace(Q) = `false THEN` the requester is $\neg$AccessControl(A).

4 PPO in-use: Implementing *MyPrivacyManager*

This section presents *MyPrivacyManager*[7], a privacy preference manager for the Social Semantic Web. It was developed to validate PPO and the formal model, i.e. to implement the creation of privacy preferences for RDF data described using PPO, and make sure the preferences are applied when requesting information, to filter requested data. Although *MyPrivacyManager* is designed to work with any Social Semantic Data[8], we will focus on defining privacy preferences for FOAF profiles. With FOAF profiles, our aim is to illustrate how the formal model can be applied to create privacy preferences and how personal information can be filtered based on such preferences.

Figure 2 illustrates the *MyPrivacyManager* architecture, which contains: (1) WebID Authenticator: handles user sign-on using the FOAF+SSL protocol; (2) RDF Data Retriever and Parser: retrieves and parses RDF data such as FOAF profiles from WebID URIs; (3) Privacy Preferences Creator: defines privacy preferences using PPO; (4) Privacy Preferences Enforcer: queries the RDF data store to retrieve and enforce privacy preferences; (5) User Interface: provides users the environment whereby they can create privacy preferences and to view other user's filtered FOAF profiles; and (6) RDF Data store: an ARC2[9] RDF data store to store the privacy preferences[10]. The implementation and functionality of these modules are explained in more detail in this section.

MyPrivacyManager employs the federated approach whereby everyone has his/her own instance of *MyPrivacyManager*. As opposed to the majority of Social Web applications which are centralised environments whereby the companies offering such services have the sole authority to control all user's data, this federated approach ensures that everyone is in control of their privacy preferences [1]. Moreover, users can deploy their instances of *MyPrivacyManager* on whichever server they prefer.

[7] Screencast online – http://vmuss13.deri.ie/myprivacymanager/screencast/screencast.html
[8] Consists of Social Web data formatted in RDF or any other structured format
[9] ARC2 — `http://arc.semsol.org`
[10] Although ARC2 was used for the implementation of MyPrivacyManager, any RDF store can be used.

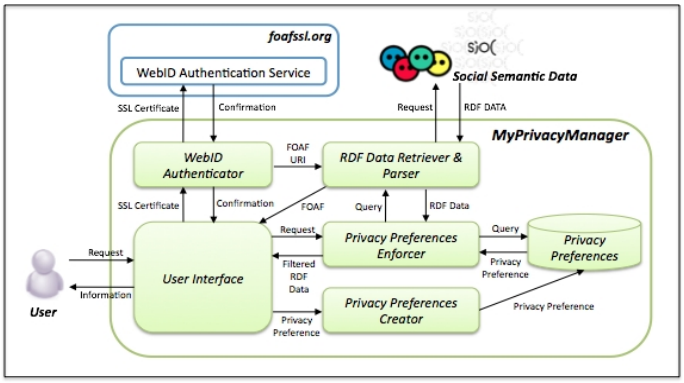

Fig. 2. MyPrivacyManager Architecture

4.1 Authentication with the WebID protocol

The WebID protocol [12] provides a mechanism whereby users can authenticate using FOAF and SSL certificates.

The WebID protocol implemented in *MyPrivacyManager* uses the libraries provided by foaf.me[11] which calls the WebID authentication mechanism offered by the FOAF+SSL Identity Provider Service[12]. This provides a secure delegated authentication service that returns back the WebID URI of the user which links to the FOAF document of the user signing in. If the identity service does not return back the WebID, then it means that the authentication has failed.

Once the user is authenticated, *MyPrivacyManager* matches the WebID URI with the WebID URI of the owner of that instance. If the owner is signed in, then the interface provides options where the user can create privacy preferences. On the other hand, if the user signed in is a requester, then the FOAF profile of the owner of that particular instance is requested. The *Privacy Preferences Enforcer* module is called (described later in this section) to filter the FOAF profile according to the privacy preferences specified by the owner of that instance.

4.2 Creating Privacy Preferences

MyPrivacyManager provides users an interface to create privacy preferences for their Social Semantic Data. The interface displays (1) the profile attributes extracted from the user's FOAF profile which the user can specify what to share in the first column and (2) other attributes (extracted from the user profile) in the second column for the user to specify who can access the specific shared information; – as illustrated in the screenshot in figure 3.

The system provides profile attributes (extracted from the user's profile) which the user can share classified as follows: (1) Basic Information consisting

[11] foaf.me — `http://foaf.me/`
[12] foafssl.org — `http://foafssl.org/`

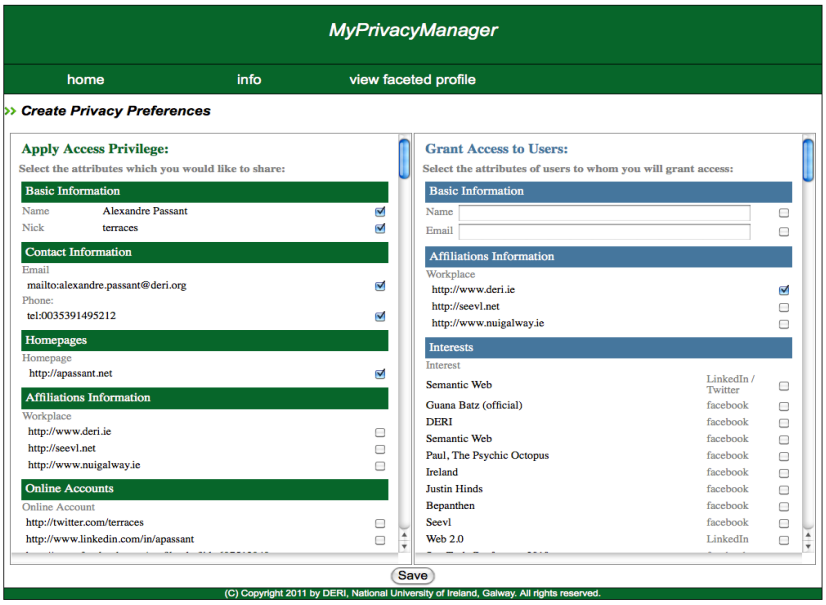

Fig. 3. The interface for creating privacy preferences in MyPrivacyManager

```
PREFIX ppo: <http://vocab.deri.ie/ppo#> .
PREFIX ex: <http://vmuss13.deri.ie/> .
ex:preference1 a ppo:PrivacyPreference;
 foaf:maker <http://foaf.me/ppm_usera#me>;
 dc:title "Restricting access to my personal information";
 dc:created "2011-06-01T13:32:59+02:00";
 ppo:appliesToStatement :Statement1;
 :Statement1
  rdf:subject <http://vmuss13.deri.ie/foafprofiles/terraces#
      me> ;
  rdf:predicate <http://xmlns.com/foaf/0.1/name>;
  rdf:object "Alexandre Passant";
 ppo:appliesToStatement :Statement2;
 :Statement2
  rdf:subject <http://vmuss13.deri.ie/foafprofiles/terraces#
      me> ;
  rdf:predicate <http://xmlns.com/foaf/0.1/nick>;
  rdf:object "terraces" ;
ppo:assignAccess acl:Read;
ppo:hasAccessSpace [
  ppo:hasAccessQuery
    "ASK { ?x foaf:workplaceHomepage <http://www.deri.ie> }"
       ].
```

Fig. 4. Privacy Preference described using PPO created in *MyPrivacyManager*

of the name, age, birthday and gender; (2) Contact Information consisting of email and phone number; (3) Homepages; (4) Affiliations consisting of the website of the user's work place; (5) Online Accounts such as Twitter, LinkedIn and Facebook user pages; (6) Education that contains the user's educational achievements and from which institute such achievements where obtained; (7) Experiences consisting of job experiences which include job title and organisation; and (8) Interests which contain a list of user interests ranked according to the calculated weight of each interest.

The attributes, extracted from the FOAF profile, which the user can select which to whom to share information must have are categorised as follow: (1) Basic Information containing fields to insert the name and email address of specific users; (2) Affiliations to share information with work colleagues; and (3) Interests to share information with users having the same interests.

Once the user selects which information to share and to whom, he/she clicks on the save button for the system to generate automatically the privacy preference using PPO. Figure 4 illustrates an example of a privacy preference described using PPO and created from *MyPrivacyManager* that restricts access to a person's name and nick name to those users who are work colleagues. Although reification is used, we intend to use named graphs in order to reduce the number of statements.

4.3 Requesting and Enforcing Privacy Preferences

MyPrivacyManager provides users to view other people's FOAF profile based on privacy preferences by logging into third party's instance. On the contrary of common Social Networks which are public by default, *MyPrivacyManager* enforces a private by default policy. This means that if no privacy preferences are set for a profile or for specific information, then this is not granted access to be viewed. In the near future, *MyPrivacyManager* will be modified to provide a feature where users can select which default setting they wish to enforce – public or private.

The sequence in which privacy preferences are requested and enforced is performed as illustrated in figure 5 which consists of: (1) a requester authenticates to another user's MyPrivacyManager instance using the WebID protocol and the system automatically requests the other user's FOAF profile; (2) the privacy preferences of the requested user's FOAF profile are queried to identify which preference applies; (3) the access space preferences are matched according to the requester's profile to test what the requester can access; (4) the requested information (in this case, FOAF data) is retrieved based on what can be accessed; and (5) the requester is provided with the data he/she can access.

MyPrivacyManager handles each privacy preference separately since each preference may contain different access spaces. Once the system retrieves the privacy preferences, for each preference it tests the access space queries with the requester's FOAF profile. If the access space query on the requester's FOAF profile returns true, then the privacy preference is considered, however, if it returns false, then that particular privacy preference is ignored. Since the access

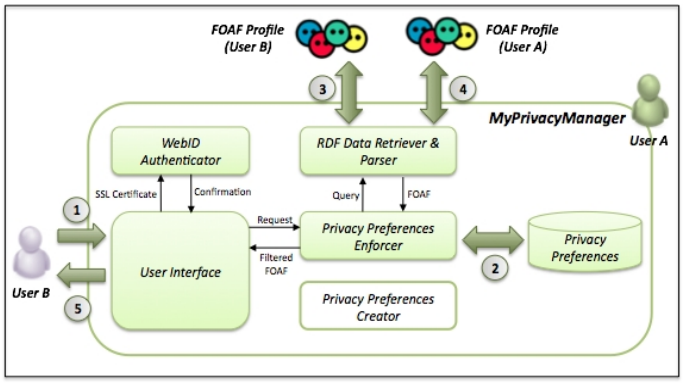

Fig. 5. The sequence of requesting third party FOAF profiles

space can contain more than one access query, in the case when one access query returns true and the other false, then by default the system enforces that the access space is true. The system then processes the restrictions and conditions defined in the privacy preference.

The system will formulate the restrictions and conditions as a group graph pattern. This group graph pattern from each privacy preference will be used to create a SPARQL query and the result from this query will be the filtered FOAF profile that can be accessed by the requester. The group graph pattern constructed from each privacy preference are combined using the keyword `UNION` within the same SPARQL query. Once the SPARQL queries are formalised, the access control privilege is assigned to the user. However, currently the system only accepts the `acl:Read` property since its purpose is to view the filtered FOAF documents of other users.

5 Related Work

The Web Access Control (WAC) vocabulary[13] describes access control privileges for RDF data. This vocabulary defines the `Read` and `Write` access control privileges (for reading or updating data) as well as the `Control` privilege to grant access to modify the access control lists (ACL). This vocabulary is designed to specify access control to the full RDF document rather than specifying access control properties to specific data contained within the RDF document. As pointed out in [9], the authors observe that protecting data does not merely mean granting access or not to the full RDF data but in most cases, users require more fine-grained privacy preferences that define access privileges to specific data. Therefore, fine-grained privacy preferences applied to RDF data using our solution create a mechanism to filter and provide customised RDF data views that only show the specific data which is granted access.

[13] WAC — `http://www.w3.org/ns/auth/acl`

The authors in [8] propose a privacy preference formal model consisting of relationships between objects and subjects. Objects consist of resources and actions, whereas subjects are those roles that are allowed to perform the action on the resource. Since the privacy settings based on this formal model combine objects and actions together, this requires the user to define the same action each time with different objects rather than having actions separate from objects. Thus, this method results in defining redundant privacy preferences. Moreover, the proposed formal model relies on specifying precisely who can access the resource. Our approach provides a more flexible solution which requires the user to specify attributes which the requester must satisfy.

The authors in [3] propose an access control framework for Social Networks by specifying privacy rules using the Semantic Web Rule Language (SWRL) [14]. This approach is also based on specifying who can access which resource. Moreover, this approach relies that the system contains a SWRL reasoner. In [5] the authors propose a relational based access control model called `RelBac` which provides a formal model based on relationships amongst communities and resources. This approach also requires to specifically define who can access the resource(s).

In [11] the authors propose a method to direct messages, such as microblog posts in SMOB, to specific users according to their online status. The authors also propose the idea of a `SharingSpace` which represents the persons or group of persons who can access the messages. The authors also describe that a `SharingSpace` can be a dynamic group constructed using a SPARQL `CONSTRUCT` query. However, the proposed ontology only allows relating the messages to a pre-constructed group.

In [7] the authors propose a system whereby users can set access control to RDF documents. The access controls are described using the Web Access Control vocabulary by specifying who can access which RDF document. Authentication to this system is achieved using the WebID protocol [12] which provides a secure connection to a user's personal information stored in a FOAF profile [6]. This protocol uses FOAF+SSL techniques whereby a user provides a certificate which contains a URL that denotes the user's FOAF profile. The public key from the FOAF profile and the public key contained in the certificate which the user provides are matched to allow or disallow access. Our approach extends the Web Access Control vocabulary to provide more fine-grained access control to the data rather than to the whole RDF document.

6 Conclusion and Future Work

In this paper we presented a formalisation of the PPO that can be used as a model whilst creating privacy preferences for any structured data. Since structured data can be used easily by other platforms taking advantage of Semantic Web technologies, privacy preferences described using the PPO can be utilised by

[14] SWRL — `http://www.w3.org/Submission/SWRL/`

any system that implements the formal model. Moreover we presented *MyPrivacyManager* which implemented the formal model of PPO in order to demonstrate how to create privacy preferences for Social Semantic Data, primarily focusing on user profiles described using FOAF. *MyPrivacyManager* also demonstrates how data is filtered on the basis of these privacy preferences.

Similar to all prototype systems, further enhancements is required to enrich *MyPrivacyManager*. It will be extended to demonstrate how data from current Social Networks such as Facebook can be filtered based on privacy preferences defined in PPO. Furthermore, since *MyPrivacyManager* assumes that the requester's information is trustworthy, the system will be extended to incorporate methodologies on how to assert the trustworthiness of requesters.

References

1. C. Au Yeung, I. Liccardi, K. Lu, O. Seneviratne, and T. Berners-Lee. Decentralization: The Future of Online Social Networking. In *Proceedings of the W3C Workshop on the Future of Social Networking Position Papers,'08*, 2008.
2. D. Boyn and E. Hargittai. Facebook privacy settings. Who cares? *First Monday*, 15(8), August 2010.
3. B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. A Semantic Web Based Framework for Social Network Access Control. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT '09*, 2009.
4. Carroll, Jeremy J. and Bizer, Christian and Hayes, Pat and Stickler, Patrick. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web, WWW'05*, 2005.
5. F. Giunchiglia, R. Zhang, and B. Crispo. Ontology Driven Community Access Control. *Trust and Privacy on the Social and Semantic Web, SPOT'09*, 2009.
6. B. Heitmann, J. Kim, A. Passant, C. Hayes, and H. Kim. An Architecture for Privacy-Enabled User Profile Portability on the Web of Data. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec '10*, 2010.
7. J. Hollenbach and J. Presbrey. Using RDF Metadata to Enable Access Control on the Social Semantic Web. In *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge, CK'09*, 2009.
8. P. Kärger and W. Siberski. Guarding a Walled Garden Semantic Privacy Preferences for the Social Web. *The Semantic Web: Research and Applications*, 2010.
9. A. Passant, P. Kärger, M. Hausenblas, D. Olmedilla, A. Polleres, and S. Decker. Enabling Trust and Privacy on the Social Web. In *W3C Workshop on the Future of Social Networking)*, 2009.
10. O. Sacco and A. Passant. A Privacy Preference Ontology (PPO) for Linked Data. In *Proceedings of the Linked Data on the Web Workshop, LDOW2011*, 2011.
11. M. Stankovic, A. Passant, and P. Laublet. Directing status messages to their audience in online communities. In *Proceedings of the 5th International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems*, 2010.
12. H. Story, B. Harbulot, I. Jacobi, and M. Jones. FOAF + SSL : RESTful Authentication for the Social Web. *Semantic Web Conference*, 2009.