PPO & PPM 2.0: Extending the Privacy Preference Framework to provide finer-grained access control for the Web of Data*

Owen Sacco
Digital Enterprise Research Institute
National University of Ireland
Galway, Ireland
owen.sacco@deri.org

John G. Breslin
Digital Enterprise Research Institute
National University of Ireland
Galway, Ireland
john.breslin@nuigalway.ie

ABSTRACT

Web of Data applications provide users with the means to easily publish their personal information on the Web. However, this information is publicly accessible and users cannot control how to disclose their personal information. Protecting personal information is deemed important in use cases such as controlling access to sensitive personal information on the Social Semantic Web or even in Linked Open Government Data. The Privacy Preference Ontology (PPO) can be used to define fine-grained privacy preferences to control access to personal information and the Privacy Preference Manager (PPM) can be used to enforce such preferences to determine which specific parts of information can be granted access. However, PPO and PPM require further extensions to create more control when granting access to sensitive data; such as more flexible granularity for defining privacy preferences. In this paper, we (1) extend PPO with new classes and properties to define further fine-grained privacy preferences; (2) provide a new light-weight vocabulary, called the Privacy Preference Manager Ontology (PPMO), to define characteristics about privacy preference managers; and (3) present an extension to PPM to enable further control when publishing and sharing personal information based on the extended PPO and the new vocabulary PPMO. Moreover, the PPM is extended to provide filtering data over SPARQL endpoints.

Categories and Subject Descriptors

1.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Method; K.4.1 [**Public Policy Issues**]: Privacy; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Design, Security

Keywords

PPO, PPM, PPMO, Access Control, Privacy, Web of Data

1. INTRODUCTION

The Web of Data encourages data creators to publish their data in standard machine processable formats so that other datasets can be linked to this published data. The importance of publishing data using open standards has also been realised by governments which have commenced to publish their data as Linked Open Government Data which enables the combination of data from different sources.

However, when publishing sensitive information, for instance personal citizen data, raises privacy concerns which discourages governments to publish personal data. This also has been experienced in Social Networks which do not provide sufficient privacy settings to conceal sensitive data [2].

We have solved these privacy shortcomings with the Privacy Preference Framework consisting of the Privacy Preference Ontology (PPO)[1] and the Privacy Preference Manager (PPM). This framework provides fine-grained, attribute based, access control vocabulary to model privacy preferences to any structured data and to grant access to specific data segments based on these preferences. However, the Privacy Preference Framework requires further enhancements to provide more flexibility and granularity when controlling access to data.

In this paper, we: (1) extend the PPO; (2) extend the Web Access Control (WAC) Vocabulary[2] to provide further access control privilege types; (3) present a new light-weight vocabulary called the Privacy Preference Manager Ontology (PPMO)[3] that provides classes and properties to define characteristics about privacy preference managers such as defining who the administrators are and grant access control privileges to modify privacy preferences; and (4) extend the Privacy Preference Manager (PPM) to support the extended PPO, the extended WAC, the new PPMO and also to provide access control to data extracted from SPARQL endpoints.

*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.

I-SEMANTICS 2012 8th Int. Conf. on Semantic Systems, Sept. 5-7, 2012, Graz, Austria

[1]PPO – `http://vocab.deri.ie/ppo#`
[2]The extended WAC vocabulary is included in the PPO
[3]PPMO – `http://vocab.deri.ie/ppmo#`

The remainder of this paper is organised as follows: Section 2 presents some scenarios where access control for the Web of Data is a concern. In Section 3, we provide an overview of the PPO and we present the extended PPO together with the formal model; the extended WAC vocabulary; and provide some examples how privacy preferences are defined using the extended PPO. In section 4 we present the new PPMO and the formal model for this vocabulary. Section 5 describes the implementation of the extended Privacy Preference Manager. Section 6 discusses related work and Section 7 presents future work and concludes the paper.

2. MOTIVATIONS

Linked Open Government Data consists of published RDF datasets containing information about how the government works and how policies are made. This knowledge is formatted in open standards so that other datasets can take advantage and link to this knowledge. However, there are a number of government datasets which are not published due to the sensitive nature of its form; these include personal records such as tax payers records and patient health records. Imagine a system where users would be able to access their personal records collected by the government which are linked to public datasets but the users records would not be publicly accessible. Moreover, the user will have the authority to control who can access the information or even authorise other users to act on his/her behalf for instance a user might authorise a relative to manage his/her medical records. Preferably, the system would provide users to specify attributes which other users must satisfy in order to be granted access to their personal records; rather than having to maintain user lists. This would be achieved by executing a SPARQL `ASK` query on the requester's profile to test whether s/he satisfies certain attributes. Furthermore, the system must provide mechanisms to specify different access control rules on different datasets since some are public by default and others require specific fine-grained privacy settings.

Another scenario would consist of linking different personal information from various domains, such as Social Networks with Governmental Data and with Financial Data. Users would not want third party users to access sensitive information but would want to specify various access control privileges to various users. For instance, users would grant a financial advisor access to specific financial data and to specific tax records but would not grant him/her access to social or medical information. Moreover, friends would not be granted access to any of the governmental or financial data but would only be allowed to access specific parts of the social data. These can be achieved by specifying attributes in SPARQL `ASK` queries that third-parties accessing the user's data must satisfy. Moreover, this also requires specifying which datasets are effected even though different datasets might contain information about the same person.

3. EXTENDING PPO - THE PRIVACY PREFERENCE ONTOLOGY

3.1 Overview

The Privacy Preference Ontology (PPO) [9], [10], [11] is a light-weight vocabulary that provides (semi-) structured data creators to describe fine-grained, attribute based, privacy preferences for granting or restricting access to specific data. PPO can be used for instance to grant access to specific sensitive information found in Linked Government Data to users that work in a particular department or for example to grant access to a part of a user's blog only to users that have similar interests. PPO provides a machine-readable way to define access control criteria such as "Grant read access to health investment costs to users that work in the government department for health" and also "Grant write access to my blog only to DERI colleagues".

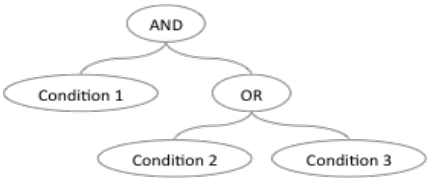

Figure 2: Nested Logical Operators

Considering that our model targets Semantic Web data or any (semi-) structured data, a privacy preference defines: (1) which resource, statement or named graph to grant access to; (2) the conditions to refine what to grant; (3) the access control privilege type; and (4) a SPARQL query, known as an `AccessSpace` containing a graph pattern representing what a user requesting information must conform. The access control type is defined by using the Web Access Control (WAC)[4] vocabulary which defines the `Read` and `Write` access control privileges (for reading or updating data).

However, the PPO lacks classes and properties that provide further granularity and flexibility when defining privacy preferences. We therefore have extended the PPO ontology with the following classes and properties (illustrated as shaded or in bold in fig. 1):

- `ppo:appliesToDataset` and `ppo:appliesToContext:` specifies which `void:Dataset` [1] or `context` (i.e. the source specified in N-Quads[5]) respectively which a privacy preference applies to. For instance, a resource's triples could exist in various data sources and hence, users can grant access to triples stored in one data source but deny access to triples of the same resource residing in another data source.

- `ppo:hasNoAccess:` defines an access control privilege that denies access and it is the inverse of `ppo:hasAccess` (and vice-versa). Previously, the PPO assumed that by default, all data is private and users have to specify which data should be granted access defined by using `ppo:hasAccess`. Forthwith, users can define whatever access control privilege they require irrespective of whether the data is public or private by default.

- `ppo:ConditionOperator:` a class that provides logical operators, defined by `ppo:Operator` (see fig. 3), consisting of conjunction, disjunction and negation. The operators can also have nested operators, defined by `ppo:hasChildConditionOperator`, that caters for connecting conditions within the same privacy preference in a tree-like hierarchy; for example (fig. 2): `condition 1 and (condition 2 or condition 3)`.

[4]WAC — `http://www.w3.org/ns/auth/acl`
[5]As specified in `http://sw.deri.org/2008/07/n-quads/`

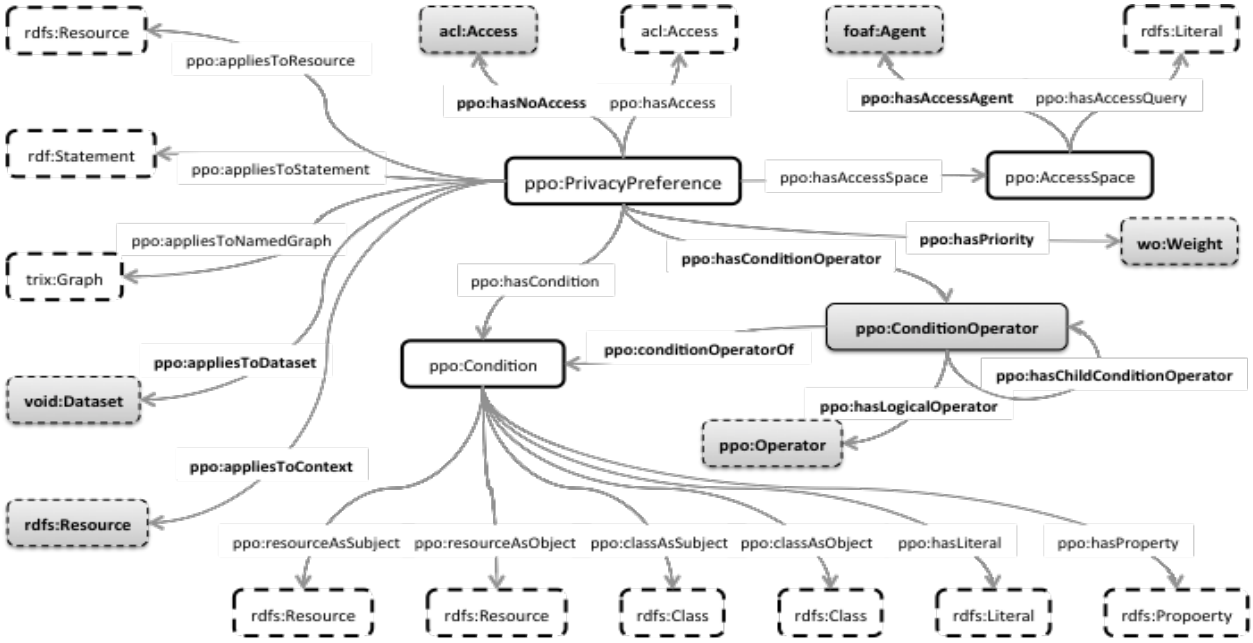

Figure 1: The Privacy Preference Ontology (PPO)

- `ppo:hasPriority`: an optional weighted value, described using the Weighting Ontology[6], that denotes the rank of a privacy preference. The privacy preference manager defines the priority scale which is used as a measure to rank the privacy preferences based on their priority value. Higher priority (ranked) privacy preferences surpass lower prioritised privacy preferences which might also solve conflicts amongst privacy preferences when they apply to the same resource, statement or named graph.

- `ppo:hasAccessAgent`: specifies an agent who should be granted (or denied) the access control privileges. Although it is recommended to use `ppo:AccessSpace` to determine who can be granted (or denied) the access control privileges since it caters for dynamic data, there are instances when users would want to grant (or deny) access to a specific agent without the need to test whether the agent satisfies specific attributes.

We also have extended the Web Access Control (WAC) vocabulary (fig. 4) to distinguish between different `write` privileges since `acl:Write` does not distinguish between an `update` and a `delete`. Hence, we have added `ppo:Update` and `ppo:Delete` as subclasses of `acl:Write` so that data creators may distinguish these access rights in the event for instance they do not want to grant a `delete` right. However, if an `acl:Write` is assigned, then both the `ppo:Update` and `ppo:Delete` rights are propagated and granted. We have also added a `ppo:Create` class for granting `create` access rights that allow users to create data elements.

3.2 A Formal Model for the Extended PPO

Following the PPO formal model specified in [10], in this section we provide the formal model of the additional classes and properties.

[6]The Weighting Ontology – `http://purl.org/ontology/wo/core#`

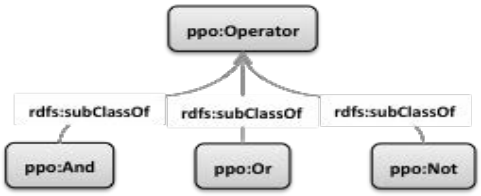

Figure 3: PPO Operators

3.2.1 *Definition 1: Data Sources*

A `privacy preference` applies to a *Dataset* or a *Context*, where:

- A *Dataset* (instance of `void:Dataset`[7]) is identified by a URI that denotes the source of a set of triples;

- A *Context* consists of a URI that denotes the source of a graph; being an instance of `rdfs:Resource`[8].

Let St be a statement, Ct a context, D a dataset and A an access control privilege. Let $Context(St, Ct)$ mean that St is contained within the context of Ct, $Dataset(St, D)$ mean that St is contained within a dataset D and $AssignAccess(Ct, A)$ or $AssignAccess(D, A)$ mean that A is assigned to Ct or D respectively.

Assigning an access privilege to a context is defined as follows:

$$\forall St(AssignAccess(Ct,A) \wedge Context(St,Ct) \Rightarrow AssignAccess(St,A)) \quad (1)$$

[7]Vocabulary of Interlinked Datasets (VoID) – `http://rdfs.org/ns/void#`

[8]Including literals

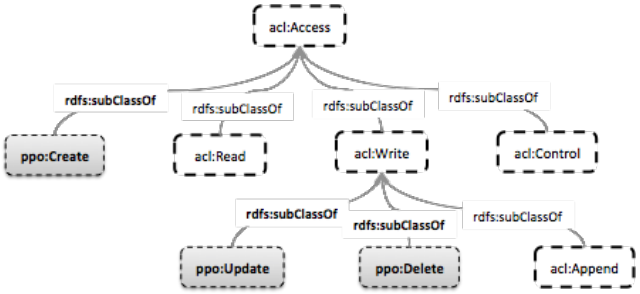

Figure 4: Extended Web Access Control Vocabulary

In other words, assigning an access privilege to a Context assigns an access privilege to all statements within that Context.

Assigning an access privilege to a dataset is defined as follows:

$$\forall St(AssignAccess(D,A) \wedge Dataset(St,D) \Rightarrow AssignAccess(St,A)) \quad (2)$$

In other words, assigning an access privilege to a Dataset assigns an access privilege to all statements within that Dataset.

3.2.2 Definition 2: Condition Operators

A `condition operator` defines how a logical operator (fig. 3) connects conditions. The `condition operator` also supports nested operators (fig. 2) which provides more flexibility and granularity when connecting conditions. The logical operators consist of:

- a logical conjuction $\wedge$ defined using `ppo:And`;
- a logical disjunction $\vee$ defined using `ppo:Or`;
- a logical negation $\neg$ defined using `ppo:Not`.

Let St be a statement, Cn a condition, Co a set of conditions in the form $Co = \{Cn_1 \wedge ... \wedge Cn_n\}$ (conjuction) or $Co = \{Cn_1 \vee ... \vee Cn_n\}$ (disjunction) or $Co = \{\neg Cn\}$ (negation) or a combination of conjunction, disjunction and negation, and A an access control privilege. Let $Condition(St, Cn)$ mean that Cn is the condition of St, $ConditionOperator(Cn, Co)$ mean that Cn is contained within Co, and AssignAccess(Co,A) mean that A is assigned to Co.

The condition operator is defined as follows:

$$\forall St(AssignAccess(Co,A) \wedge (Condition(St,Cn) \wedge \neg Condition(St,\neg Cn)) \wedge ConditionOperator(Cn,Co) \Rightarrow AssignAccess(St,A)) \quad (3)$$

In other words, assigning an access privilege to a Condition Operator assigns an access privilege to all statements related to the conditions within that Condition Operator and not to those statements that are in negated conditions within that Condition Operator.

3.2.3 Definition 3: Access Control Privilege

An `access control privilege` defines the `create`, `read` and/or, `write` privileges which also includes `update` and/or `delete` privileges that can be specified either separately or globally (by assigning the `write` privilege). Hence, the `access control privilege` is defined as:

$$AccessControl = \{create,read,write,update,delete\} \quad (4)$$

3.2.4 Definition 4: Access Agent

An `Access Agent` can either be a person, an organisation, a group, a software or a physical artefact as defined by `foaf:Agent`. An access agent is defined within an `access space` which can have multiple access space queries and multiple access agents. Therefore, an access space can be defined as the set:

$$AccessSpace = \{\{accessquery_1,...,accessquery_n\} \wedge \{accessagent_1,...,accessagent_n\}\} \quad (5)$$

3.3 Creating Privacy Preferences

In this section, we provide some examples of privacy preferences created using the extended PPO.

The first example defines a privacy preference which is (1) applied to all triples of a resource of a particular investment cost type; (2) applied to a particular dataset; (3) that must have the resource URI as subject; (4) the object as an IT system type; (5) has access control privileges `read` and `update`; and (6) is granted access to all those that work at HHS.

```
ex:pp1
  a ppo:PrivacyPreference;

  ppo:appliesToResource
    <http://www.example.org/Investment
        /90000001>;

  ppo:appliesToDataset
    <http://www.example.org/repositories/
        dataset1>;

  ppo:hasConditionOperator [
    ppo:conditionOperatorOf [
      ppo:resourceAsSubject
        <http://www.example.org/Investment
            /90000001>];

    ppo:hasLogicalOperator ppo:And;

    ppo:conditionOperatorOf [
      ppo:resourceAsObject
        <http://www.example.org/ITSystem
            /8000000002>]
  ];

  ppo:hasAccess acl:Read;
  ppo:hasAccess ppo:Update;

  ppo:hasAccessSpace [
      ppo:hasAccessQuery
      "ASK { ?x foaf:workplaceHomepage
          <http://www.hhs.gov> }"].
```

The second example defines a privacy preference that utilises the nested logical operators. The privacy preference (1) applies to a `foaf:Person`'s resource URI; (2) must contain a property `foaf:givenName` and `foaf:familyName` and `(foaf:mbox or foaf:homepage)`; (3) has access control privileges `read`; and (4) is granted access to all those that work at DERI.

```
ex:pp2
  a ppo:PrivacyPreference;
```

```
ppo:appliesToResource
  <http://vmuss13.deri.ie/userprofiles/
     winu#me>;

ppo:hasConditionOperator [
  ppo:hasLogicalOperator ppo:And;

  ppo:conditionOperatorOf
    [ppo:property foaf:givenName];

  ppo:conditionOperatorOf
    [ppo:property foaf:familyName];

  ppo:hasChildConditionOperator [
    ppo:hasLogicalOperator ppo:Or;

    ppo:conditionOperatorOf
      [ppo:property foaf:mbox];

    ppo:conditionOperatorOf
      [ppo:property foaf:homepage]]
];

ppo:hasAccess acl:Read;

ppo:hasAccessSpace [
    ppo:hasAccessQuery
    "ASK { ?x foaf:workplaceHomepage
        <http://www.deri.ie> }"].
```

4. THE PRIVACY PREFERENCE MANAGER ONTOLOGY (PPMO)

In [10] we developed a Privacy Preference Manager (PPM) that assists users to create privacy preferences described using the PPO. The manager also provides a filtering module that enforces the privacy preferences and filters the data when requested. The manager requires several configuration settings, for example specifying who the owner is, that conform to the data creator's needs. However, there is no standard way in defining these configuration settings which therefore requires the user to configure each manager in order to utilise the privacy preferences. Moreover, the manager assumed that only the owner of the data has privileges to create privacy preferences, but with government data or enterprise data for instance, administrators are required to control access to the data. Therefore, we have created the Privacy Preference Manager Ontology (PPMO) that provides a light-weight vocabulary to define attributes about administering the privacy preference manager and also describes several configuration properties including who can control the privacy preferences stored within the privacy preference manager and also to define default values to solve conflicts amongst privacy preferences.

4.1 Ontology

The Privacy Preference Manager Ontology (PPMO) illustrated in figure 5 provides: (1) a main class called `PrivacyPreferenceManager` that identifies a privacy preference manager; (2) a property that defines the owner of the manager; (3) some properties that define administration including which access control privilege is granted to administrators and which attribute patterns that users must satisfy to have administrator rights; (4) some properties that define which default access control privileges should be assigned in cases when the data do not fall under any privacy preference; (5) and some properties that define which default access control privileges should be assigned in cases when there are conflicts between privacy preferences.

The classes and properties provided by the PPMO are explained below.

- `ppmo:hasOwner`: defines the owner of the privacy preference manager.
- `ppmo:Administration`: is a class that provides classes and properties that specify administration attributes. This class provides properties to specify what access control privilege an administrator has over the privacy preferences stored within the manager. The access control privilege are defined using the `ppmo:hasAdminAccess` and `ppmo:hasAdminNoAccess` which grant and /or deny the access type described using the extended Web Access Control (WAC) vocabulary (fig. 4) - for creating, reading, updating and deleting privacy preferences. The `acl:Control` can be used to define who can modify the privacy preference manager's configuration settings described using PPMO. This class also provides a `ppmo:AdminSpace` class that defines who the administrators are. The `ppmo:hasAdminSpaceQuery` specifies a SPARQL query that tests whether a user satisfies certain attributes to be an administrator; for instance a SPARQL `ASK` query would test whether the user works in the IT department and is in the group called "Admin". The `ppmo:hasAdministrator` property defines statically a specific person, group, organisation, software or other physical entity that is an administrator. It is recommended to use `ppmo:hasAdminSpaceQuery` since it gives the advantage of not having to maintain administrator lists as it caters for dynamic data, for example a particular person who is no longer in the "Admin" group; whereas the `ppmo:hasAdministrator` is useful to define administrators that do not change frequent, such as the owner of the manager.
- `ppmo:hasDefaultAccess` and `ppmo:hasDefaultNoAccess`: define the default access privileges which the manager grants and/or denies in the case when resources, statements or named graphs do not fall under any privacy preference whilst enforcing the privacy preferences to filter the RDF data. Moreover, these properties are the inverse of each other.
- `ppmo:hasDefaultConflictAccess` and `ppmo:hasDefaultConflictNoAccess`: define the default access privilege which the manager grants and/or denies in the case when conflicts arise amongst privacy preferences. Conflicts occur when resources, statements or named graphs fall under more than one privacy preference. Moreover, these properties are the inverse of each other.
- `ppmo:hasPriorityScale`: defines the default priority scale which the privacy preference manager uses to rank the privacy preferences. Based on this scale, the higher prioritised privacy preferences are enforced first and overrule lower prioritised privacy preferences. Hence, if a resource, statement or named graph falls under more than one privacy preference, the higher prioritised access privilege is granted or denied. If more than one privacy preference have the same priority value and they apply to the same resource, statement or named graph, then the default conflict access privilege is applied.

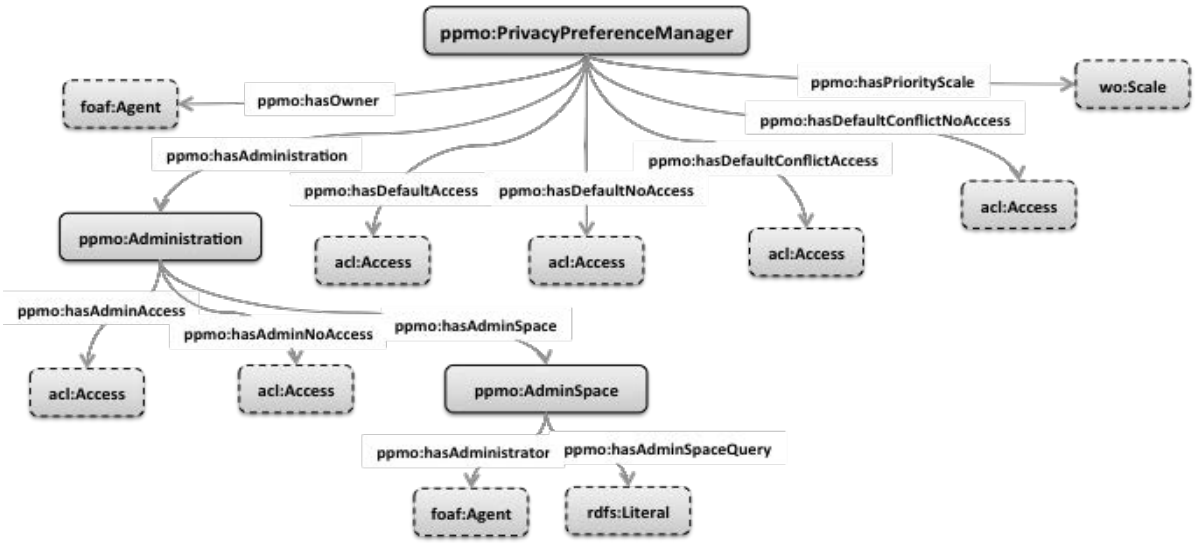

Figure 5: The Privacy Preference Manager Ontology (PPMO)

4.2 A Formal Model for the PPMO

In this section we provide the formal model of the classes and properties of the PPMO.

4.2.1 Definition 1: Owner

An `Owner` can either be a person, an organisation, a group, a software or a physical artefact denoted by a WebID [13] and there can only be one owner; which is defined as:

$$Owner = \{WebID\} \quad (6)$$

4.2.2 Definition 2: Administration

Administration consists of (1) the `access control privileges` which can be granted and/or denied to administrators; and (2) the `admin space` which defines who the administrators are.

An `access control privilege` defines the `create`, `read` and/or, `write` (which also includes `update` and/or `delete` privileges that can be specified either separately or globally by assigning the `write` privilege) privileges for creating, reading, updating, deleting privacy preferences; and the `control` privilege to maintain the privacy preference manager's configuration settings described using PPMO. Hence, the `access control privilege` is defined as:

$$AccessControl = \{create,read,write,update,delete,control\} \quad (7)$$

The `AdminSpace` defines who the administrators are by either using `admin space queries` to test whether users satisfy specific attributes to have administration privileges or by defining specific administrators. The admin space is defined as follows:

$$AdminSpace = \{\{adminspacequery_1,...,adminspacequery_n\} \wedge \{administrator_1,...,administrator_n\}\} \quad (8)$$

4.2.3 Definition 3: Default Access Control Privileges

PPMO provides default access control privileges for the resources, statements or named graphs which are not covered by any privacy preference; and also default access control privileges for when conflicts amongst privacy preferences occur during the filtering of RDF data. The defaults access control privilege is defined using the the extended Web Access Control (WAC) vocabulary (figure 4) and is defined as follows:

$$DefaultAccessControl = \{create,read,write,update,delete\} \quad (9)$$

4.3 Creating Configuration Files using PPMO

Configuration settings for a Privacy Preference Manager can easily be created using the PPMO and the extended Web Access Control (WAC) vocabulary (figure 4). For example a user wants to create the following settings: (1) owner has WebID: `http://vmuss13.deri.ie/userprofiles/winu#me`; (2) administrators must satisfy a SPARQL `ASK` query that tests if the user has the admin email address[9]; (3) administrators are granted `create, read` and `update` access privileges but are denied `delete` and `control` privileges; (4) the privacy preference manager grants default access `read` and denies default access `create` and `write`; (5) the privacy preference manager grants default conflict access `read`; and (6) the privacy preference manager uses a priority scale from 0 to 1. This configuration is specified as follows:

```
ex:config1
  a ppmo:PrivacyPreferenceManager;

 ppmo:hasOwner
  <http://vmuss13.deri.ie/userprofiles/
      winu#me>;

ppmo:hasAdministration [
  ppmo:hasAdminAccess ppo:Create;
  ppmo:hasAdminAccess acl:Read;
  ppmo:hasAdminAccess ppo:Update;
  ppmo:hasAdminNoAccess ppo:Delete;
  ppmo:hasAdminNoAccess acl:Control;

  ppmo:hasAdminSpace [
    ppmo:hasAdminSpaceQuery
    "ASK { ?x foaf:mbox
```

[9]We assume that a PPO / PPMO interpreter would know the common prefixes for SPARQL queries, while they could also be defined in the ASK pattern.

```
      <mailto:admin@example.org>}" ;

    ppmo:hasAdministrator
      <http://vmuss13.deri.ie/userprofiles
          /winu#me>]];

ppmo:hasDefaultAccess acl:Read;
ppmo:hasDefaultNoAccess ppo:Create;
ppmo:hasDefaultNoAccess acl:Write;

ppmo:hasDefaultConflictAcces acl:Read;

ppmo:hasPriorityScale [
  wo:max_weight "1.0";
  wo:min_weight "0.0";].
```

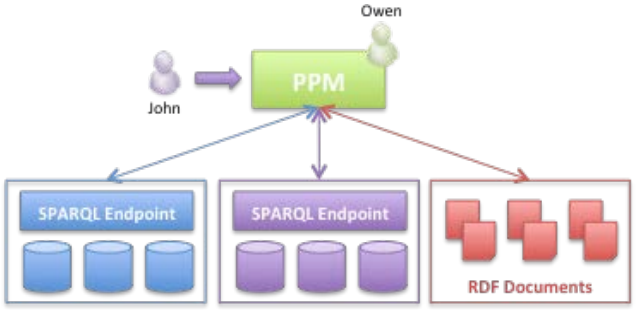

Figure 6: PPM Architectures

5. EXTENDING PPM - THE PRIVACY PREFERENCE MANAGER

The Privacy Preference Manager[10], implemented in [10], is a Web application that provides users the facility to create privacy preferences on structured data and also it filters data based on these privacy preferences when third-parties request user's data. In this section, we explain how we have extended the Privacy Preference Manager (PPM) to take into account the extended PPO, extended WAC vocabulary and the new PPMO vocabulary. Moreover, we have extended the PPM to support SPARQL endpoints (as illustrated in figure 6).

5.1 Architecture

The architecture provides users to (1) authenticate to a PPM instance using the WebID protocol and if they are administrators, they can create privacy preferences on the data; and (2) authenticate to a PPM instance and if they are not administrators, they can request data either through SPARQL queries (which are encrypted and sent securely to the SPARQL endpoint) or request RDF documents, and the result of the request is filtered by the PPM based on the privacy preferences.

5.2 Authentication

After the user authenticates successfully using the WebID protocol, the PPM has been extended to use the configuration settings described using PPMO to check whether the user is an administrator or not. If the user is an administrator, based on the configuration settings, the user is granted administration rights which might include creating privacy preferences. If the user is not an administrator, then the user is presented with the option to request data.

5.3 Creating Privacy Preferences

The PPM was extended so that privacy preferences can be created for data residing in SPARQL endpoints. The user can specify the SPARQL endpoint location and the SPARQL query to retrieve the data on which the user wants to create the privacy preferences. When the PPM retrieves the data, the triples are previewed and the user can select the various PPO and WAC properties (including the extended classes / properties) from drop-down boxes. Once the user completes the privacy preference and clicks on the create button, the PPM will automatically generate the privacy preference described using the extended PPO and WAC vocabularies.

5.4 Filtering Data

Filtering data has been extended to filter data residing in SPARQL endpoints. The user is presented with a screen to enter the SPARQL endpoint location and the SPARQL query. The PPM then retrieves the whole result set and filters the data based on the privacy preferences stored in the PPM. The strategy which the PPM uses to filter the data is the following: (1) PPM retrieves all the privacy preferences; (2) maps the triples to the privacy preferences which they fall under; (3) creates a list of the mapped triples and another list of triples which are not mapped; (4) for the mapped triples, the PPM executes the access space queries on the requester's profile - if successful then the has access and the has no access privileges are assigned to the requester, otherwise, the triples are added to the not mapped list of triples; (5) the PPM's default has access privileges and has no access privileges are assigned to the not mapped triples; (6) the triples are presented to the user based on the assigned access privileges.

6. RELATED WORK

The Web Access Control (WAC) vocabulary[11] describes access control privileges for RDF data – the `Read` and `Write` access control privileges (for reading or updating data), and the `Control` privilege for granting access to modify the access control lists (ACL). This vocabulary is designed to specify access control to the full RDF document rather than to specific data contained within the RDF document.

The Platform for Privacy Preferences (P3P)[12] specifies a protocol that enables Web sites to share their privacy policies with Web users expressed in XML. This platform does not ensure that Web sites act according to their publicised policies and since this platform aims to enable Web sites to define their privacy policies, it does not solve our aim of enabling users to define their own privacy preferences.

The authors in [7] 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. The proposed formal model relies on specifying precisely who can access the resource and therefore, our approach provides a more flexible solution since users

[10]Screencast online – `http://vmuss13.deri.ie/ppmv2/screencast/screencast.html`

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

[12]P3P — `http://www.w3.org/TR/P3P/`

specify attributes which requesters must satisfy.

The authors in [8] propose an access control model in semantic networks whereby users define their policies to resources to predefined users or user groups. This model only works in "closed world" environments whereby everything is private by default unless specified otherwise.

The authors [14] propose a similar access control vocabulary and manager that uses SPARQL queries to test requesters whether they satisfy specific attributes. However, their model applies only to named graphs, unlike our model which we apply to statements and resources; hence providing finer-grained access control. Moreover, this model does not provide properties to specify which specific dataset to apply the rules; does not provide nested logical operators and does not provide the negation operator.

The authors in [3] propose an access control framework for Social Networks by specifying privacy rules using the Semantic Web Rule Language (SWRL)[13]. This approach is also based on specifying who can access which resource. In [4] 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 [12] 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 [6] 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 [13] which provides a secure connection to a user's personal information stored in a FOAF profile [5]. 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.

7. CONCLUSION AND FUTURE WORK

In this paper we extended the PPO with new classes and properties to provide more flexible and finer-grained privacy preferences. Moreover, we have extended the WAC vocabulary with new classes to define more specific access control types. We also presented the new PPMO vocabulary that allows data owners to describe characteristics of privacy preference managers, including specifying administration rights. Moreover, we have extended the Privacy Preference Manager to cater for the extended PPO, extended WAC, the new PPMO vocabulary and also to support filtering on data residing in SPARQL endpoints.

Similar to all prototype systems, further enhancements is required to enrich the Privacy Preference Manager. It will be extended to assert the trustworthiness of requester's information since it currently assumes that the requester's information is trustworthy.

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

8. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets on the design and usage of void, the "vocabulary of interlinked datasets". In *Proceedings of the Linked Data on the Web Workshop, LDOW2009*, 2009.

[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] F. Giunchiglia, R. Zhang, and B. Crispo. Ontology Driven Community Access Control. *Trust and Privacy on the Social and Semantic Web, SPOT'09*, 2009.

[5] 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.

[6] 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.

[7] P. Kärger and W. Siberski. Guarding a Walled Garden Semantic Privacy Preferences for the Social Web. *The Semantic Web: Research and Applications*, 2010.

[8] T. Ryutov, T. Kichkaylo, and R. Neches. Access control policies for semantic networks. In *Proceedings of the 2009 IEEE International Symposium on Policies for Distributed Systems and Networks*, POLICY '09, pages 150–157, Washington, DC, USA, 2009. IEEE Computer Society.

[9] O. Sacco and A. Passant. A Privacy Preference Ontology (PPO) for Linked Data. In *Proceedings of the Linked Data on the Web Workshop, LDOW2011.*

[10] O. Sacco and A. Passant. A Privacy Preference Manager for the Social Semantic Web. In *Proceedings of the 2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation, SPIM2011*, 2011.

[11] O. Sacco, A. Passant, and S. Decker. An Access Control Framework for the Web of Data. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-11*, 2011.

[12] 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.

[13] H. Story, B. Harbulot, I. Jacobi, and M. Jones. FOAF + SSL : RESTful Authentication for the Social Web. *Semantic Web Conference*, 2009.

[14] S. Villata, N. Delaforge, F. Gandon, and A. Gyrard. Social semantic web access control. In *Procs of the 4th International Workshop Social Data on the Web, SDoW2011*, 2011.