Privacy Aware and Faceted User-Profile Management Using Social Data [1]

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Owen Sacco [a], Fabrizio Orlandi [a] and Alexandre Passant [a]

[a] *Digital Enterprise Research Institute,*
National University of Ireland, Galway
Ireland
E-mail: firstname.lastname@deri.org

Abstract. 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. Recent studies showed that privacy in Social Networks is a major concern when user profiles are publicly shared, revealing that most users are aware of privacy settings. 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. Yet, the studies show that users require more complex privacy settings as current systems do not meet their requirements. Hence, we propose a platform-independent system that allows end-users to set fine-grained privacy preferences for the creation of privacy-aware faceted user profiles on the Social Web.

Keywords: Social Web, Privacy, User Profiling, Semantic Web

1. Introduction

Social networks, using non structured data formats, provide minimum privacy settings such as granting privileges to all people belonging to one's social graph to access his/her information [9]. We envisage a social network using structured data which provides users to specify which information can be accessed by specific users who have for instance similar attributes (*e.g.* interests, contact information, etc.). This would make users feel more confident when publishing online their information, especially since they specifically know who can access their information. Although applications are being developed to export user information from closed social networks into structured data such as 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 [32]. Additionally, most social networks have the sole authority of controlling all user's data [5]. Therefore, a system that provides users to create fine-grained privacy preferences which can be used by different platforms is required. This system will provide users to be fully in control of who can access their personal information and who can access their published structured data.

Moreover, the possibility to share personal profile information across heterogeneous Social Web systems would provide several benefits to end-users in terms of personalisation and recommendations. Aggregation and reuse of user profiles across Web systems would

[1] 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.

provide more accurate and comprehensive profiles [2]. This would provide already available personal information even when a new user joins a new service (the so-called "cold-start" problem) [28], and it would increase the precision of the recommendations. On the other hand an important drawback is that typically users would not want to share their entire user profile with all their Web services or all their contacts. Therefore, a system providing users full control over their complete personal information letting them define different *facets* of their user profile would be beneficial.

The system we propose and describe in this article aims at providing a user the necessary tools and options for setting fine-grained privacy preferences on his/her full private profile which is the result of the aggregation of different distributed profiles from different sources. As displayed in Figure 1, the system we implemented is composed of two main parts: the user profiling module, and the Privacy Preference Manager – *MyPrivacyManager*. The first part, as described in Section 3, is the component that collects profile data from different social media websites (*e.g.* personal information, activities, interests, etc.), generates specific user profiles for each platform, and then merge them in a global complete user profile. The second component (*MyPrivacymanager*) described in Section **??**, allows the owner of the full user profile to specify his/her privacy preferences on the profile. It also manages the requests of other users by asking for the requester's profile information: it replies with a faceted, or filtered, user profile which is the result of the privacy preferences applied to the full profile based on the profile information of the requester.

This article is structured as follows. In Section 2 we review the related work for the research areas of user profiling and privacy. Then, in Section 3, details about the architecture implemented for user profiling on the Social Web and integration of user models are provided. Section 4 describes the Privacy Preference Ontology (PPO), implemented in *MyPrivacyManager*, which is a light-weight vocabulary to describe privacy preferences for restricting (or granting) access to specific data. Then, a user study that motivated our work and provided feedback and guidelines for the implementation of the system is presented in Section 5. The description of the implementation itself is in Section **??**, and an evaluation of the system is provided in Section 7. Section 8 provides some concluding remarks and our future work.

2. Related Work

2.1. User Profiles

During the last years we have assisted to the growth of Web applications using or collecting data on their users and their behaviour in order to provide adapted and personalized contents and services. This caused the need for exchange, reuse, and integration of their data and user models. A new research challenge then emerged, seeking solutions for user modelling and personalisation across application boundaries [56][15]. In this section we review the current state of the art for the research fields of user modelling and personalisation especially in relation with the Semantic Web.

2.1.1. Introduction to User Modelling

User modelling techniques are applied by adaptive web systems to represent with formal models the interests, knowledge and goals of their users. These user models are then necessary to provide a personalized experience for different users, for instance by filtering the content relevant to the user on a website, rearranging elements on a page, or recommending users with similar interests. Approaches for personalisation cannot be applied without an accurate understanding of the user. The field of user modelling [19][35] is then focused on techniques for the description of user knowledge into user models which constitute the basis for adaptive systems.

According to Brusilovsky et al. [12], web personalisation now constitutes a large research field that includes communities such as web science, hypertext, user modelling, machine learning, information retrieval, intelligent tutoring systems, cognitive science, Web-based education, etc. In this section we focus our analysis in particular on user modelling and personalisation techniques for adaptive web systems. These techniques can be grouped in three areas [13]: personalisation of information retrieval, personalisation of browsing and personalisation through filtering and recommendation. Web systems for adaptive recommendation are a more specific type of adaptive web systems which attempt to deduce the users goals and interests from his/her browsing activity and recommend a list of related content and relevant links to the user [13]. The core of this field is represented by the *user modelling* research area. Personalisation and adaptation are based on complex information related to user's knowledge, activities, interests, social relations, etc. that could be modelled using structured representa-

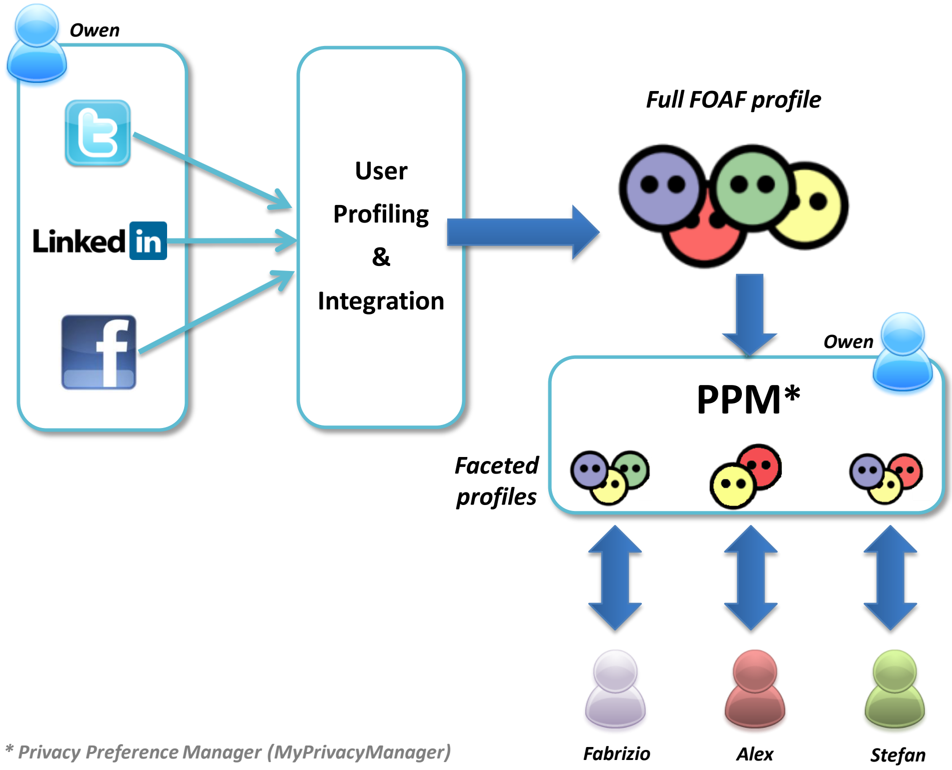

Owen
Linked in
User Profiling & Integration
Full FOAF profile
Owen
PPM*
Faceted profiles
* Privacy Preference Manager (MyPrivacyManager)
Fabrizio
Alex
Stefan

Fig. 1. Overview of the entire system architecture

tions and user modelling techniques. Generalising, this information about a user is typically stored and represented with a *user profile*.

As regards the user modelling field, at the moment three main *challenges* are commonly identified. The *first* one is about how to retrieve information about user interests, knowledge, behaviour and social context. In other words the challenge is to find ways to collect all the useful information needed to build user profiles. The *second* important aspect is related to how to manage and represent user models in an interoperable and scalable way. Hence the goal is the aggregation and exchange of user models between heterogeneous applications, and the accurate representation of complete and global user profiles. The *third* challenge regards the use of Semantic Web technologies and the Web of Data in order to enrich user models and provide an interoperable and more accurate representation of user profiles.

2.1.2. *User Information Retrieval*

As regards the work done on user information retrieval, latest techniques to track the user behaviour have to cope with the current highly dynamic and socially interactive web applications and have to be extended to collect fine-grained data from user interactions to provide better information for adaptive systems. Additionally, the collected data must be managed in ontologies to share user behaviour information with other adaptive systems. Zhou et al. [57] focus on mining client-side access logs of a single user or client and then incorporate fuzzy logic to generate a usage ontology. Schmidt et al. [47] embed concepts into a portal which provide the context for JavaScript events, which are collected and used to adjust the portal. All the relevant user interface elements are linked to a concept ontology containing semantic information about the elements. None of these approaches make full use of semantic technologies. First steps in the direction of semantic technologies are done but they still cannot be applied across heterogeneous applications and lack the necessary extensibility and dynamism (see Section 2.1.4).

Szomszor et al. [51] investigate the idea of merging users' distributed tag clouds to build richer profile ontologies of interests, using the FOAF vocabulary and matching concepts to Wikipedia categories. The authors experimented with over 1,300 users who showed high activities in both of the two websites Del.icio.us[1] and Flickr[2]. For each user, data about each of the two tag clouds has been retrieved and then merged. The results described in the paper show that, on average, 15 new concepts of interest were learnt for each user when expanding tag analysis to their tag cloud in the other folksonomy. In this case the user profiles generated are represented using popular lightweight vocabularies such as FOAF.

In this context is also very relevant the *user identification* aspect. In order for applications to share information about users, mechanisms for the identification of users are necessary. Identity-based protocols such as OpenID[3] or WebID[4] can be used for users to link their different identities on the Web. Google Friend Connect[5] provides an API which exemplifies the use of OpenID and OAuth to integrate registered users, existing login systems, and existing data with new social data and activities. It is based on open standards (OpenID, OAuth and Open Social[6]) and allows users to control and share their data with different social websites. Moreover, the WebFinger[7] protocol documents a way to get a XML file describing how to find a user's public metadata from that user's email address like identifier, providing then information about existing user accounts linked to that user.

2.1.3. Architectures for User Model Interoperability

Latest developments on user modelling involve interoperability and portability of user models [15][56]. The rapid growth of user adaptive and social systems collecting information about users led to the replication of user data over many applications. This inevitably conducted researchers to deal with an important challenge: user model interoperability, or in other words the process of exchanging distributed and heterogeneous user data across applications [55]. User model interoperability would provide several advantages under quantitative and qualitative aspects. It allows the collection of more data and more accurate data about users, the acquisition of user modelling functionalities that systems do not themselves implement, a solution to the well-known "cold start" problem during the user model initialization phase and a consequent speed-up of this phase. On the other hand achieving interoperability on the Web, a completely open and dynamic environment, is a complex and challenging task that requires open and agreed standards and a high level of alignment of the involved systems.

In the context of user model representation and management increasing relevance is attributed to the interoperability of the representations. Applications typically store their user information in a proprietary format. This leads to a distributed web model of a user with several partial user models in different applications potentially duplicating information. Therefore, the challenge is to solve the heterogeneity of the user models. Current research on user model management and aggregation emphasizes two different strategies [37]. The first strategy introduced in [6] uses a generic user model mediation framework with the goal of improving the quality of recommendations. The actual UM mediation in the framework is done by specialized mediator components which translate the data between different models using inference and reasoning mechanisms. In their subsequent work [7] Berkovsky et al. still focus on cross-representation mediation of user models describing its practical implementation and evaluating the outcome of the collaborative to content-based filtering user model mediation. As they state in their paper "the mediation procedure allows bootstrapping the empty UMs and enriching the existing UMs in a content-based recommender system, and, as a result, more accurate recommendations are generated". The second strategy focuses on the standardization of user models to allow data sharing between applications. Heckmann [27] proposes an ontological approach, the General User Model Ontology (GUMO), as a top level ontology for user models and suggests the ontology to be the standard model for user modelling tasks. Another standardization approach is to define a centralized user modelling system that is used and updated by all connected applications [36]. The drawback of the mediation layer approach is the effort needed to aggregate such heterogeneous user models, while standardized user models suffer from the lack of a common standard.

Different solutions and architectures have been proposed in order to solve the interoperability problem and can be categorized in three types of approaches: centralized, decentralized and mixed. This categorization is mainly based on two factors: the *physical storage*, or where the user data is maintained, and the *conceptualization* of the model, that is "how the

[1] `http://www.delicious.com/`

[2] `http://www.flickr.com`

[3] `http://openid.net`

[4] `http://www.w3.org/wiki/WebID`

[5] `http://code.google.com/apis/friendconnect/`

[6] `http://code.google.com/apis/opensocial/`

[7] `http://code.google.com/p/webfinger/`

user model component is conceived in terms of being shared or not between systems"[15]. A *centralized* approach then represents systems that are both physically and conceptually centralized, *decentralized* approaches are physically and conceptually distributed, and *mixed* approaches refer to systems that are physically decentralized and conceptually centralized. Another possible classification, in terms of architecture, is based on *standardization-based* and *mediation-based* approaches [56]. The former aims at defining a standard whom all the involved applications have to comply, the latter aims at transferring and adapting user modelling data from one system to another. In most of the cases standardization-based approaches are conceptually centralized and mediation-based ones are conceptually decentralized. To note that so far user modelling systems are evolving from centralized to decentralized architectures. This tendency is motivated mainly by the difficulties in developing and adopting a unique and common user modelling standard and by the intrinsic decentralized nature of the Web. Moreover centralized systems are by definition affected by the single point of failure problem and by the privacy and security of users' information which is all stored in a single point. A comprehensive list of user modelling systems appropriately categorized is provided by two recently published surveys by Carmagnola et al.[15] and Viviani et al.[56]. We refer to these two publications for a complete overview of the state of the art in this research field. Here we select and describe only the work that we found particularly relevant in our research context.

Latest developments and examples of centralized architectures are described in PersonisAD [4], UMS (User Modelling Server) [34], MUMS (Massive User Modelling System) [11].

Assad et al. [4] developed a framework called *PersonisAD* that aims at supporting the development of context-aware applications using distributed user models. The framework is targeted at ubiquitous applications and supports the management of different kinds of models, such as models of users, places, sensors, services and devices. Therefore not only user data is exchanged, but also data about the environment. However an application complying with this framework has to use a common user model at an ontological level of the components in the environment in order to have knowledge about the components themselves and about the different contexts in which the user models are organized.

Kobsa and Fink [34] (see also [33] and [21]) developed a *User Modelling Server (UMS)* based on the *Lightweight Directory Access Protocol (LDAP)*. It allows external applications to submit and retrieve information about users whose models are represented in the system. Therefore it provides a user modelling service to other applications and is capable of representing different types of models, from user profiles to system and service models. The type of exchanged data is strictly related to users (demographic data, interests and preferences) and application usage.

Brooks et al. [11] in their work describe the *MUMS* system, a *Massive User Modelling System*. It is a centralized system that provides a user modelling/adaptation service, it supports "the just-in-time *production, delivery* and *storage* of user modelling information". It is suitable for describing any domain that can be expressed in RDF/OWL, hence uses Semantic Web techniques and standards. In order to represent the users it adopts a shared user model ontology and all the managed information is expressed in RDF. The interaction between the user data producers and user modellers systems utilizing the data is mediated by a central broker component, while the architecture and the communication layer is Web service based.

As regard decentralized approaches, in [39] Metha et al. propose a standardization-based approach using a common ontology-based user context model (UUCM — Unified User Context Model) as a basis for the exchange of user profiles between multiple systems. Cross system personalisation is then obtained relying on an unified profile for each user which is stored inside a "Context Passport"[43]. Further developments of this work by Metha et al. are described in [38] where the authors propose machine learning techniques for automatically matching user models. Dependencies between profiles are computed analysing data provided by users sharing their profile across different systems and learning from that population. The UUCM is also encoded as RDF Schema augmented with OWL expressions enabling exchange possibilities with other Semantic Web enabled systems.

Another example of decentralized architectures for user modelling is presented by Heckmann et al. [25] where user-adaptive systems exchange user information using UserML [24], a RDF-based user model exchange language, and the *General User Model Ontology* (GUMO) [26], an ontology for the uniform interpretation of decentralized user models. This is another example of a standardization-based approach as the GUMO ontology is proposed as the uniform interpretation of distributed user models in Semantic Web environments. It is so far the most comprehensive user modelling ontology but at the same time it is very extensive and it might be complex to implement in a real system. Moreover this vocabulary has to be adopted by the systems that want to exchange user models, so an *a priori* agreement between the systems is necessary, in the same way as in [39] previously described.

In [16] [14] a new approach for user model interoperability is proposed. The authors propose a framework that ”deals with semantic heterogeneity of user models and automates the user model exchange across applications“. It is inspired by Semantic Web technologies and represents an intermediate solution which combines both a flexible user model representation and an automatic semantic mapping of user data across different systems. An algorithm based on evidential reasoning has beed developed in order to create mappings between concepts and values present in different user models and measure their similarity (*Object Similarity Algorithm* and *Property Similarity Algorithm*). User models are represented and exchanged in RDF and queried using SeRQL (Sesame RDF Query Language)[8].

2.1.4. *Semantic Web Technologies for User Modelling*

Interesting research that bridges the gaps between user information retrieval/profiling and the Semantic Web has been presented by Szomszor et al. [51]. The authors investigate the idea of merging users' distributed tag clouds to build richer profile ontologies of interests, using the FOAF vocabulary and matching concepts to Wikipedia categories. We previously described this work in Section 2.1.2 and it is particularly relevant that the authors demonstrate the benefits of the amalgamation of multiple Web2.0 user-tagging histories in building personal semantically-enriched profiles of interest. The user profiles generated are also represented using a popular lightweight vocabulary such as FOAF.

A survey on adaptive systems adopting Semantic Web technologies is provided in [53]. The author describes a classification of adaptive systems based on a distinction between *strong semantic techniques* and *weak semantic techniques*. The first regard systems based on the Semantic Web approach and the latter regard technologies that basically aim at annotating resources in order to enrich their meaning. The survey is mainly focused on weak semantic approaches, these are particularly successful in contributing to user modelling tasks especially when combined with social tagging features. On the other hand strong semantic techniques are more suitable for user knowledge integration and reasoning. The authors also suggest that a category of mixed approaches is growing and it benefits of the advantages of both the technologies in different tasks. The analyzed tasks belongs to the topics of domain modelling and management, context modelling and management, adaptation, personalisation and privacy. The authors provide a matrix summarizing the reviewed systems on the basis of the semantic technology that was used and the task it was used for.

Relevant related work on Semantic Web applied to user modelling and personalisation has been done by Aroyo et al. [3]. In this work the authors highlight the challenges they see in the near future for user modelling and the adaptive semantic web and a review of the research in this field is provided. In the state of the art review the authors analyze the differences between past user modelling solutions (in traditional ”closed world“ Web-based or application-based systems) and new research on ”open“ and Semantic Web based solutions. The fundamental tasks identified by the authors that contribute to user modelling are: user identification, user property representation, and sharing adaptation functionalities. An analysis of some of the possible solutions to these tasks is provided by the authors, and relevant related work is also presented. Moreover the authors provide a set of challenges on this research field describing possible future developments and scientific questions. The major question in user identification investigates how to identify a person on the Web, her multiple identities across different applications and what are the trust and privacy aspects involved. As regards user knowledge the main challenge is to find ways to share user models, and this implies the definition of common vocabularies and interoperable representations of objects and values of user properties. Finally in their work Aroyo et al. highlight an important aspect about the openness of the Web of Data and the related implications of this on users' experience: an open approach to user knowledge would produce different new use cases and knowledge management approaches, especially users should then be able to inspect and edit their own data. Related and more practical work by the same authors and others is described in [48] and [54] where, as part of the No-Tube project, by using the Linked Data cloud, semantics can be exploited to find complex relations between

[8] http://www.openrdf.org/doc/sesame/users/ch06.html

the user's interests and background information of TV programmes, resulting in potentially interesting recommendations. Also in another paper [20] Denaux et al. present how interactive user modelling and adaptive content management on the Semantic Web can be integrated in a learning domain to deal with common adaptation problems (e.g. cold start, inaccuracy of assumptions, knowledge dynamics, etc.).

Finally in the previous section we already described the work done by Carmagnola et al. [16] [14] representing one of the most advanced user modelling systems adopting semantic technologies. The use of RDF for representing user models and the reasoning capabilities implemented with a "SPARQL-like" language (SeRQL) on top of the user models in order to obtain automatic mapping between heterogeneous concepts are the strongest points of their implementation. A drawback of their system is the lack of scrutable user models, it is not possible for a system user to consult her user model created by the application.

As we described previously, some of the systems for user model interoperability analysed use RDF or OWL to represent user models however the user models created cannot be shared or integrated easily with other different systems or on the Web of Data because of the complexity and particularity of the ontologies used. Moreover, in almost all the cases, reasoning capabilities on top of the user data are not implemented using Semantic Web technologies.

2.1.5. *A comparison of Systems for User Model Interoperability*

In this section we show a comparison table including the systems for user model interoperability that we reviewed previously in this Section 2.1. In the table we display only the systems with a complete implementation: from the information retrieval task, to the mapping of user concepts and values, to the provision of integrated user profiles or a personalisation service available to other external applications. Moreover this is not a complete table including all the applications in the state of the art, but it represents a selection of some of the most interesting systems from our perspective considering our research goals. In Table 1 we categorise the systems according to their architecture and then we list the positive and negative aspects that we see in those implementations. Some of these aspects are subjective and somehow influenced by our research background. For further details please refer to the description of the systems in this section.

2.2. *Privacy*

2.2.1. *Markup Languages*

The eXtensible Access Control Markup Language [44] is an XML based language for expressing a large variety of access control policies. XACML components consist of: (1) *Rule* - defines what the request that the policy applies to and whether it is allowed or denied; (2) *Policy* - contains a set of rules; and (3) *Policy set* - a set of policies. Although the XACML is widely used, it does not provide the necessary elements to define fine-grained access control statements for structured data. It also does not contain enough semantics to describe what the actual access restriction is about and also does not semantically define which attributes a requester must satisfy.

2.2.2. *Vocabularies*

The Web Access Control (WAC) vocabulary[9] describes access control privileges for RDF data. This vocabulary enables owners to create access control lists (ACL) that specify access privileges to the users that can access the data. The WAC 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 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 [45], 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.

In [49] the authors propose a method to direct messages, such as microblog posts in SMOB, to specific users according to their online status defined by the Online Presence Ontology (OPO). 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.

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

-	Architecture	Pros	Cons

Assad et al. 2007 ***(PersonisAD)***	Centralized;
	Standard-based;
		User + environment models;
		Scrutable user models;
			No Semantics;
			Common user model;

Kobsa & Fink 2006 ***(UMS)***	Centralized;
	Mediation-based;
		No common user model;	No Semantics;
			Based on LDAP;

Brooks et al. 2004 ***(MUMS)***	Centralized;
	Standard-based;
		Real-time service; User models in RDF/OWL;	Common user model;

Metha et al. 2005	Decentralized;
	Standard-based;
		Machine learning for model matching;
		UUCM user models in RDF/OWL;
		User + environment models;
			Common user model;

Heckmann et al. 2005	Decentralized;
	Standard-based;
		GUMO ontology in OWL;
		User + environment models;
		Scrutable user models;
			Common user model;
			Complexity of GUMO ontology;

Carmagnola et al. 2009	Decentralized;
	Mediation-based;
		User model in RDF;
		Reasoning for user model mapping;
			No scrutable user model;

Table 1

Comparison of the reviewed systems

2.2.3. *Formal Models*

The authors in [32] 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.

In [23] 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).

2.2.4. *Platforms, Protocols and Frameworks*

The Platform for Privacy Preferences (P3P)[10] specifies a protocol that enables Web sites to share their privacy policies with Web users. The privacy policies are expressed in XML which can be easily parsed by user agents. This platform does not ensure that Web sites act according to their publicised policies. Moreover, 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 Protocol for Web Description Resources (POWDER)[11] is designed to express statements that describe what a collection of RDF statements are about. The descriptions expressed using this protocol are text based and therefore do not contain any semantics that can define what the description states. Therefore, our approach enables users to define what the privacy preferences are about and hence facilitate other systems to use such preferences.

The authors in [17] propose an access control framework for Social Networks by specifying privacy rules using the Semantic Web Rule Language (SWRL) [12]. This approach is also based on specifying who can access which resource. Moreover, this approach relies that the system contains a SWRL reasoner.

The authors in [22] propose a Semantic e-Wallet aiming at supporting identification and access of personal resources for contextual-aware environments.

[10] P3P — `http://www.w3.org/TR/P3P/`

[11] POWDER — `http://www.w3.org/TR/powder-dr/`

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

Context-aware applications will only reveal credentials in the correct context.

In [31] 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 [50] which provides a secure connection to a user's personal information stored in a FOAF profile [29]. 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.

2.2.5. *Annotation Based Access Control Models*

The authors in [42] propose a tag-based model to create privacy settings for medical applications that consist of annotating resources with different access policy rules. The privacy rules are denoted in a system specific language which only the system can interpret the access control. The authors in [41] also propose an annotation based access control model. This approach enables users to annotate the resource and also to annotate users. The access control rules therefore specify which resource annotations can be accessed by which user annotations. Although this approach might be more flexible than other systems, it still relies on specifying who can access the resource.

2.2.6. *Access Control Mechanisms for RDF Repositories*

In [1] the authors propose an approach to provide a fine-grain access control mechanism for RDF repositories. This approach does not depend on the RDF store but it provides mechanisms to evaluate a query and provide a highly optimised query that enforces privacy preference policies. The authors argue that when defining a priori which subsets of an RDF store can be accessed by some requester, all requesters and their allowed graphs must be known in advance. Since in their scenario they explain to have a dynamic system whereby the data is continuously changing, they state that these methods would not hold since named graphs cannot be precomputed for each possible scenario that would result in a large number of named graphs, and also this creation process would excessively slow down the system. Therefore, based on privacy preferences, which are formatted in a textual structure, their system expand SPARQL `SELECT` or `CONSTRUCT` queries in the `FROM` part of the query with path expressions and boolean expressions before such queries are executed on the RDF data store. Hence, once the query is expanded with the necessary policies, the expanded query is then sent to the data store to be executed which will result in filtered data according to the user's privacy policies. The authors state that this method is also better than querying the RDF data and post filtering the data since this method is costly in terms of time and resources.

3. User Profiling

Users on the Social Web interact with each other, create/share content and express their interests on different social websites with many user accounts and different purposes. On each of these systems, personal information consisting of a portion of the complete profile of the user is recorded. With respect to "complete user profile", we intend to build the full set of personal information belonging to a person by aggregating the distributed partial user profiles on each Social Web system. Each partial user profile might contain the user's personal and contact information, her interests, activities and social network of contacts. These details are often used by each application for personalisation and recommendation purposes. All the distributed user profiles on the Web represent different *facets* of the user therefore their aggregation provides a more comprehensive picture of a person's profile.

Aggregation of user profiles brings several advantages [2]: it allows for information reuse across different systems, it solves the well-known "cold start" problem in personalisation/recommendation systems [28], and provides more complete information to each individual Social Web service. On the other hand aggregation and reuse of user information might lead to privacy problems. Users might want to explicitly avoid sharing some information on specific social platforms based on privacy and security concerns. This is one of the main reasons that motivates the work presented here. We provide the user the ability to create and aggregate her distributed profiles and then to set fine-grained privacy preferences on the full aggregated profile. This allows the user to gain full control on the personal information that can be shared across social platforms or with specific users.

In this section we detail how we retrieved user information from different social websites and how we created a global user profile aggregating the original distributed ones. The social websites used in the experiment are Twitter[13], LinkedIn[14] and Facebook[15]. From each one of these websites we create RDF-based user profiles using FOAF [10] and other popular lightweight vocabularies. Following in this section we provide more details about the user profiling process and the merging of the different sources of information.

3.1. Creation and Aggregation of User Profiles

In this section we provide an overview of the creation process of user profiles and the related modelling phase using popular lightweight ontologies. The extraction and generation of user profiles from social networking websites is composed of two main parts: data extraction and the subsequent generation of application-dependent user profiles. After this phase the next steps involve the representation of the user models using popular ontologies, and then, finally, the aggregation of the distributed profiles.

First, in order to collect private data about users on social websites it is necessary to have access granted to the data by the users. In social services like Twitter this is not always necessary because many users post all the microblog messages publicly. In other social media sites it is necessary to request access to the profile data in order to see most of the data which is often private by default. Then, once the authentication step is accomplished, the two most common ways to fetch the profile data is by using an API provided by the system or by parsing the Web pages.

Once the data is retrieved, in order to benefit of the potentialities offered by Semantic Web technologies, the next step is the data modelling using standard ontologies. In this case, a possible way to model profile data is to generate RDF-based profiles described using the FOAF vocabulary [10]. FOAF is one of the most popular lightweight ontologies on the Semantic Web and using this vocabulary as a basis for representing users' personal information and social relations eases the integration of heterogeneous distributed user profiles. Semantic Web technologies and standard ontologies are the main supports for the development of interoperable services, and these standards make it easier to connect distributed user profiles.

```
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
  <foaf:name>Fabrizio Orlandi</foaf:name>
  <foaf:nick>BadmotorF</foaf:nick>
  <foaf:mbox rdf:resource="mailto:fabrizio.
      orlandi@deri.org"/>
  <foaf:homepage rdf:resource="http://www.
      deri.ie/about/team/member/
      fabrizio_orlandi"/>
  <foaf:phone rdf:resource="tel
      :+35391494035"/>
  <foaf:workplaceHomepage rdf:resource="http
      ://www.deri.ie"/>
  <foaf:account>
    <sioc:UserAccount rdf:about="http://
        twitter.com/BadmotorF">
    </sioc:UserAccount>
  </foaf:account>
  [...]
</foaf:Person>
```

Fig. 2. An example of a part of a FOAF profile in RDF/XML

A FOAF profile constitutes of a FOAF PersonalProfileDocume that describes a foaf:Person: a physical person that has several properties describing her and holds online accounts on the Web. Some of the main FOAF properties describing users are [16]: name, nick, phone, homepage, mbox, etc. In Listing 2 we show an example of one of the FOAF profiles that we generated for our experiment. Apart from the basic contact information we can see in the example that the person "Fabrizio Orlandi" holds an account on Twitter and that account is represented with a SIOC UserAccount[17]. We then extend FOAF with the SIOC ontology [8] to represent more precisely an online account on the Social Web.

Additional personal information about users' affiliation, education, and job experiences can be modelled using the DOAC vocabulary[18]. This allows us to represent the past working experiences of the users and their cultural background. An example of our modelling so-

[13] http://www.twitter.com

[14] http://www.linkedin.com

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

[16] FOAF Specification: http://xmlns.com/foaf/spec/

[17] SIOC Specification: http://rdfs.org/sioc/spec/

[18] DOAC Specification: http://ramonantonio.net/doac/0.1/

```
[...]
<doac:experience>
  <doac:Experience>
    <doac:title>Research Assistant</doac:
        title>
    <doac:organization>DERI, Galway</doac:
        organization>
  </doac:Experience>
</doac:experience>
<doac:education>
  <doac:Education>
    <doac:title>M.Sc.</doac:title>
    <doac:organization>National University of
         Ireland, Galway</doac:organization>
  </doac:Education>
</doac:education>
[...]
```

Fig. 3. Modelling job experiences and education with DOAC. An example in RDF/XML.

lution for job experiences and education is showed in Listing 3.

Another important part of a user profile is represented by the user's interests. In Listing 4 we display an example of an interest about "Semantic Web" with a weight of 0.5 on a specific scale (from 0 to 1) using the Weighted Interests Vocabulary (WI)[19] and the Weighting Ontology (WO)[20].

In order to compute the weights for the interests common approaches are based on the number of occurrences of the entities, their frequency, and possibly some additional factors. These factors might depend on whether or not the interest was implicitly mined or explicitly showed by the user, or depending on a time-based function which computes the decay of the interests over time , or based on the trustworthiness of the social platform, and so on. In other words many different factors can be considered to influence the weights of the interests.

Finally, the phase that follows the modelling of the FOAF-based user profiles and the computation of the weights for the interests is the aggregation of the distributed user profiles. When merging user profiles it is necessary to avoid duplicate statements (and this is done automatically by a triplestore during the insertion of the statements). Furthermore, as in the case of the interests, if the same interest is present on two different profiles it is necessary to: represent the interest only once, recalculate its weight, and update the provenance of the interest keeping track of the source where the interest was derived from. As regards the provenance of the interest, as showed in Listing 4, we use the property `wasDerivedFrom` from the Open Provenance Model[21] (OPM) to state that the interest was originated by a specific website. In the example in Listing 4 we can observe that the interest is derived from both the Twitter and LinkedIn user accounts.

```
[...]
<foaf:topic_interest rdf:resource="http://
    dbpedia.org/resource/Semantic_Web" />
<wi:preference>
  <wi:WeightedInterest>
    <wi:topic rdf:resource="http://dbpedia.
        org/resource/Semantic_Web" />
    <rdfs:label>Semantic Web</rdfs:label>
    <wo:weight>
      <wo:Weight>
  <wo:weight_value rdf:datatype="http://www.
      w3.org/2001/XMLSchema#double">0.5</wo:
      weight_value>
  <wo:scale rdf:resource="http://example.org
      /01Scale" />
      </wo:Weight>
    </wo:weight>
    <opm:wasDerivedFrom rdf:resource="http://
        www.twitter.com/BadmotorF" />
    <opm:wasDerivedFrom rdf:resource="http://
        www.linkedin.com/in/fabriziorlandi"
        />
  </wi:WeightedInterest>
</wi:preference>
[...]
<wo:Scale rdf:about="http://example.org/01
    Scale">
  <wo:max_weight rdf:datatype="http://www.w3.
      org/2001/XMLSchema#decimal">1.0</wo:
      max_weight>
  <wo:min_weight rdf:datatype="http://www.w3.
      org/2001/XMLSchema#decimal">0.0</wo:
      min_weight>
</wo:Scale>
```

Fig. 4. Representing an interest (*Semantic Web*) and its weight (*0.5*) found in two sources (Twitter and LinkedIn)

As regards the computation of the aggregated global weight for the interest generated by multiple sources, we propose a simple generic formula that can be

[19] WI Specification: `http://purl.org/ontology/wi/core#`

[20] WO Specification: `http://purl.org/ontology/wo/core#`

[21] OPM Specification: `http://openprovenance.org/`

adopted for merging the interest values of many different sources. The formula is as follows:

$$G_i = \sum_s w_s * w_i \tag{1}$$

Where:
G_i = global weight for interest *i*
w_s = weight associated to the source *s*
w_i = weight for the interest *i* in source *s*.

Using this formula it is possible to specify static weights associated to each source depending on which source we want to give more relevance. A particular use case is explained in detail in the next section.

3.2. Use Case: Twitter, LinkedIn, Facebook

3.2.1. Experiment: Setup

For the particular use case of our experiment we considered three different social networking sites:Twitter, LinkedIn and Facebook. In order to collect user data from each of those platforms, we developed three different types of applications. For Twitter and Facebook we implemented two similar PHP scripts that make use of the respective query API publicly accessible on the Web. For LinkedIn we use a XSLT [22] script that parses the LinkedIn user profile page and generates an XML file containing all the attributes found on the page. The user information collected from Twitter is the publicly available data posted by the user, *i.e.* his/her latest 500 microblog posts. As regards Facebook and LinkedIn, since part of the user data is not exposed publicly, we requested access to the private data directly to the users. For the LinkedIn data extraction process we parsed the private profile pages of the users (containing both private and public information). As regards Facebook, our PHP script implements the OAuth 2.0 [23] authentication system (the authentication system adopted by Facebook) to access the private data of the users.

The set of users selected for the experiment is a limited number of users (15 users) and all of them have at least two user accounts on the aforementioned social websites. Only 10 of them hold all the three user accounts and we manually linked/identified their accounts on the Web. In order to identify user accounts belonging to the same person we could have used services like the Google Social Graph API [24] where users holding a Google account explicitly link their other user accounts and Web sites and make this information available. In [2] the authors show in their experiment that using the Social Graph API they were able to collect data for 338 persons who had five different user accounts. Since the purpose of this work is not to evaluate the accuracy of the generated user profiles we did not need to collect data from a large number of users. The users that we selected are the same users that, at a later stage, evaluated the implemented system.

3.2.2. Experiment: Information Extraction

On Twitter, the extraction of user information is based on entity/concept recognition on the tweets posted by the users. We analyse the latest 500 tweets of each user and we run an entity extraction algorithm to identify the objects and concepts included in the tweets, assuming that the user is interested in those “topics”. We opted for a simple dictionary-based entity extraction similar to the extraction technique used in *Twarql* [40] because of its performance. In *Twarql* this algorithm is relatively fast and used for realtime analysis of a continuous stream of microposts. A set of 3.5 million entities[25] from DBpedia is loaded as a trie (prefix tree) and longest sub-string match is performed against each post. In this way we extract entities mentioned in the tweets to statistically generate the list of concepts the user is interested in. The extracted concepts are then ranked and weighted using their frequency of occurrences. The same approach is described in [52] where the authors create user profiles for Twitter users in a similar manner.

While on Twitter we create profiles with implicitly inferred interests, on LinkedIn and Facebook we collect not only interests that have been explicitly stated by the users, but also their personal details such as contacts, workplace, education, etc. As described before, on LinkedIn we are able to get private information from our user base by parsing their private profile page using a XSLT script. The retrieved attributes include, for instance, *phone number*, *homepage*, *education*, *affiliations* and *working experiences*, *interests*, etc.

Similarly on Facebook we are able to generate a profile including some personal details of the user and her interests which have been explicitly recorded by the user by clicking the *“like”* button on the Facebook en-

[22] http://www.w3.org/TR/xslt
[23] http://oauth.net/2/
[24] http://code.google.com/apis/socialgraph/
[25] http://wiki.dbpedia.org/About (July 2011)

tities/pages. The PHP script developed authenticates our user account and checks the permissions to read private data from the other users' profiles (using the OAuth authentication protocol). The personal data is then fetched through the Facebook Graph API [26] as well as the interests (*likes*) that are then mapped to the related Facebook pages representing the entities.

We represent the entities/concepts on which the user is interested in using both DBpedia and Facebook resources. Linking the interests to the related DBpedia resources enables reuse and interoperability of the interests since DBpedia is a widely adopted and large dataset. Not all the Facebook entities can be mapped directly to DBpedia, for this reason we keep Facebook resources in our profiles to represent the interest extracted from Facebook. In our future work we plan to automatically map the Facebook entities to DBpedia as well, in order to have homogeneity between the concepts/entities and better interoperability.

3.2.3. *Experiment: Weighting the Interests*

The weights for the interests are calculated in two different ways depending on whether or not the interest has been implicitly inferred by the entity extraction algorithm (the Twitter case) or explicitly recorded by the user (the LinkedIn and Facebook cases). In the first case, the weight of the interest is calculated dividing the number of occurrences of the entity in the latest 500 tweets by the total number of entities identified in the same 500 tweets. In the second case, since the interest has been manually set by the user, we assume that the weight for that source (or social networking site) is 1 (on a scale from 0 to 1). So we give the maximum possible value to the interest if it has been explicitly set by the user.

Our approach as regards the computation of the new weights as a result of the aggregation of the profiles is straightforward. We consider every social website equally in terms of relevance, hence we multiply each of the three weights by a constant of $1/3$ (approximately 0.33) and then we sum the results. The following formula summarises the computation of a new global weight (G) as result of the three original weights (w_1, w_2, w_3). Formula 2 is the same formula that we propose in the previous section (formula 1) with the following values: $w_s = 1/3 \forall s$. Hence:

$$G_i = 1/3 * w_i + 1/3 * w_i + 1/3 * w_i \tag{2}$$

Once the full FOAF-based profiles were generated we could insert them in the privacy preference manager *MyPrivacyManager* to continue the experiment of creating faceted user profiles based on privacy preferences (see Section **??**).

4. Privacy

In section 1 we stated that current social networks lack fine-grained access control mechanisms to limit access to specific structured data to particular users. We therefore designed a light weight vocabulary called the Privacy Preference Ontology (PPO) [46]. PPO provides a light-weight vocabulary enabling Linked Data creators to describe fine-grained privacy preferences for granting access to specific data. Access is granted to users satisfying certain attributes rather than granting (or restricting) access to specific users. PPO can be used for instance to restrict part of a FOAF user profile only to users that have similar interests. PPO provides a machine-readable model to define settings such as "provide my phone number only to colleagues" or "grant write access to this picture gallery only to people I've met in real-life". In order to facilitate the creation of privacy preferences, user interfaces (as explained in section **??**) can be developed so that end users do not require to create manually the privacy preferences themselves.

4.1. *The Privacy Preference Ontology (PPO)*

The Privacy Preference Ontology (PPO), as illustrated in figure 5, provides:

(1) A class called `PrivacyPreference` for defining privacy preferences;
(2) Properties for defining access restrictions to statement(s), resource(s) and/or named graph(s);
(3) Properties for defining conditions to specify which particular statement(s), resource(s) and/or named graph(s) is being restricted;
(4) Properties for defining which access privilege should be granted; and
(5) Properties for defining attribute patterns that must be satisfied by requesters.

[26] `https://developers.facebook.com/docs/reference/api/`

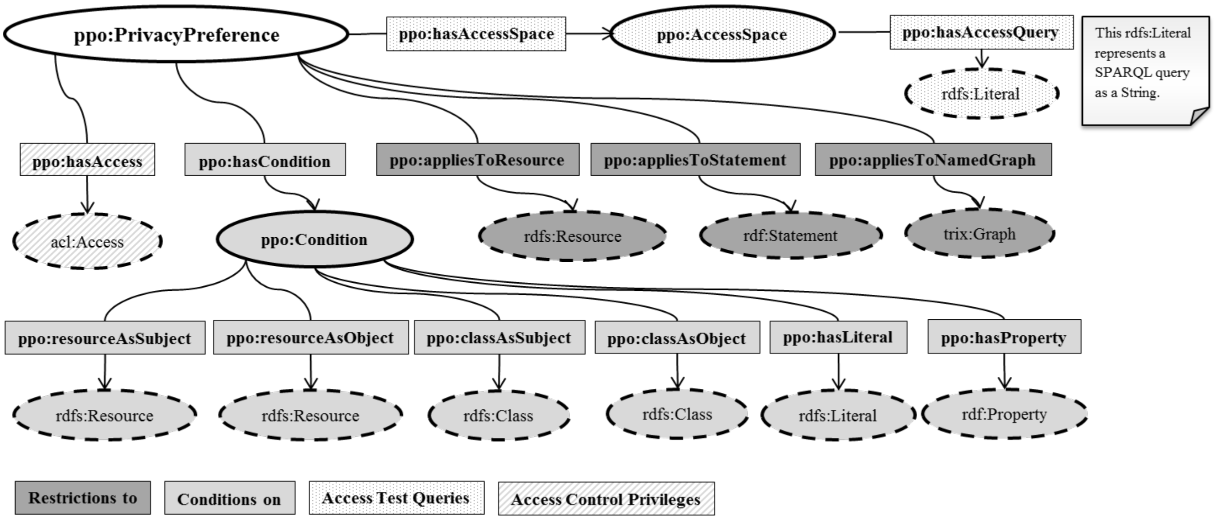


Fig. 5. A graphical representation of the Privacy Preference Ontology (PPO)

```
<http://www.example.org/pp>
  a ppo:PrivacyPreference;

  ppo:appliesToResource <http://foaf.me/
      ppm_usera#me>;
[...]
```

Fig. 6. An example of defining the `appliesToResource` property

4.1.1. Restrictions

The PPO provides various properties for defining what will the privacy preference be applied to: resources, statements and/or named graphs.

appliesToResource The `appliesToResource` property specifies which particular resource is restricted or shared. This property defines the resource's URI contained in the statements that are restricted or shared. A resource URI can be defined either as a `subject`, `predicate` or `object` within a statement and hence, `appliesToResource` property restricts any statement having the resource's URI in any part of the triple. Figure 6 shows a snippet of a privacy preference defining the `appliesToResource` to a specific person's resource URI.

appliesToStatement The `appliesToStatement` property specifies which particular statement is restricted or shared. This property requires reification; that the *subject*, *predicate* and *object* are specified of the statement which is restricted. Figure 7 shows a snippet of a privacy preference defining the `appliesToStatement` to a specific statement.

```
<http://www.example.org/pp>
  a ppo:PrivacyPreference;

  ppo:appliesToStatement :Statement1;

  :Statement1
    rdf:subject :Alice;
    rdf:predicate foaf:phone;
    rdf:object <tel:00353123456789>;
[...]
```

Fig. 7. An example of defining the `appliestoStattement` property

appliesToNamedGraph When restricting group of statements, it would be cumbersome to create a preference for each statement using the `appliesToStatement` property. Named graphs [18] can be used to combine statements which are identified by a URI. Therefore, named graphs encode structured data, such as statements, within a graph which is assigned a URI (hence, named graph). Thus, a privacy preference can be applied to a named graph's URI by using the `appliesToNamedGraph` property which is also applied to the structured data contained within the named graph. Figure 8 shows a snippet of a privacy preference defining the `appliesToNamedGraph` to a specific named graph URI.

4.1.2. Conditions

The `Condition` class provides several properties for defining circumstances in which a privacy preference may also apply. Therefore, restrictions define

```
<http://www.example.org/pp>
  a ppo:PrivacyPreference;

  ppo:appliesToNamedGraph :G1;

:G1 {
  :Alice rdf:type foaf:Person .
  :Alice foaf:Phone <tel:00353123456789> }
[...]
```

Fig. 8. An example of defining the `appliesToNamedGraph` property

which specific data is restricted and conditions define attributes which the data must have for the privacy preference to apply. Restrictions and conditions can also be defined within the same privacy preference for defining further fine-grain access control. Conditions are defined similar to the example in figure 10.

resourceAsSubject and resourceAsObject The `resourceAsSubject` property provides a condition whereby a resource's URI must be defined as a subject in a statement. Similarly, the `resourceAsObject` property applies to whenever a resource's URI is defined as an object.

classAsSubject and classAsObject The `classAsSubject` property applies to those statements that contain the instance of the class specified as the subject of the statement. Likewise, the `classAsObject` property applies to those statements that the object defines the instance of the class.

hasProperty The property `hasProperty` defines conditions for all instances of a particular predicate used within RDF statements.

hasLiteral The `hasLiteral` property defines conditions for specific literals defined as objects within RDF statements. This property is useful when the user is not aware of which property describes the literal. On the other hand, this property can also be used together with the `hasProperty` to restrict a particular value defined by a specific property.

4.1.3. Access Test Queries

PPO provides access test query properties which define to whom access is granted.

hasAccessQuery Since it is cumbersome to update manually user control lists that specify who is granted (or restricted) access due to interests or relationships changing over time, access queries can be used to test whether a requester satisfies a set of attributes. These access queries are SPARQL `ASK` queries that contain a graph pattern specifying which attributes and properties must be satisfied. The SPARQL query is described as a `Literal` in the privacy preferences using the `hasAccessQuery` property. The `hasAccessQuery` property is defined within a class called `AccessSpace` which denotes a space of access test queries. Moreover, there are instances whereby users would require to construct dynamic user groups which can also be achieved by a SPARQL `CONSTRUCT` query defined with the `hasAccessQuery` property.

```
  ppo:hasAccessSpace
    [   ppo:hasAccessQuery
      "ASK { ?x foaf:topic_interest
          <http://dbpedia.org/resource/
               Semantic_Web> }"].
```

Fig. 9. An example of defining an access space

hasAccessSpace The property `hasAccessSpace` represents the relationship between the privacy preference and the access space.

4.1.4. Access Control Privileges

hasAccess The `hasAccess` property defines the access control described using the Web Access Control vocabulary described in section **??**. This vocabulary provides `Read` and `Write` access control privileges used in PPO.

4.2. Creating Privacy Preferences

Privacy preferences can easily be created using the PPO and the Web Access Control vocabulary. For example if a user wants to create a privacy preference that restricts the phone number to whoever works at DERI, this is illustrated in figure 10[27].

This example illustrates that wherever in the user's profile there is a statement that contains a property `foaf:phone` then all statements containing this property are restricted. If the user requires a particular `foaf:phone` to be restricted, then the user must also define the phone number in the condition by using the `hasLiteral` property. The SPARQL query defined by the `ppo:hasAccessQuery` is executed on the requester's FOAF profile by the system. The query returns either `true` or `false` whether the requester's

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

```
<http://www.example.org/pp1>
  a ppo:PrivacyPreference;

  ppo:hasCondition
    [ ppo:hasProperty foaf:phone ];

  ppo:hasAccess acl:Read;

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

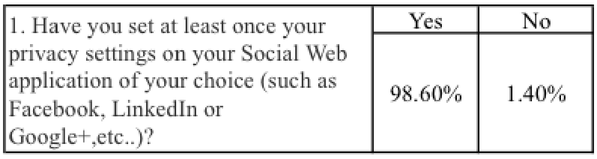
Fig. 10. An example of a privacy preference described using PPO

information satisfies the graph pattern or not. If the query returns `true` then the requester is granted the access control privilege to the statement, otherwise the requester is not granted the access privilege.

So far are assuming that the requester is a trustworthy source, however, in the near future we will focus on identifying trustworthiness of requesters and sources. Moreover, we are assuming that person has authority to create privacy preferences for the dataset. However, algorithms such as [30] can be applied to identify whether or not the person has the authority to create the privacy preferences on a particular dataset.

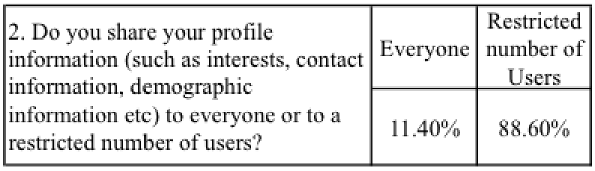
5. Preliminary User Study

Prior to implementing the privacy preference manager that provides users to create privacy preferences for generating faceted profiles, we first conducted an online survey in order to understand what users think about protecting their personal information published online. This survey also serves as the requirements for designing our interface; to know which options to provide to end-users. The survey contains 7 questions which, together with the results from 70 users, are illustrated in figures 11 - 17.



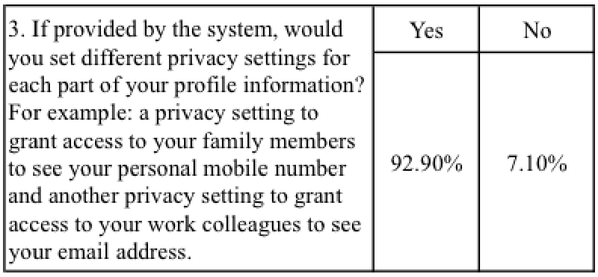
1. Have you set at least once your privacy settings on your Social Web application of your choice (such as Facebook, LinkedIn or Google+,etc..)?	Yes	No
	98.60%	1.40%

Fig. 11. User Study - Question 1



2. Do you share your profile information (such as interests, contact information, demographic information etc) to everyone or to a restricted number of users?	Everyone	Restricted number of Users
	11.40%	88.60%

Fig. 12. User Study - Question 2

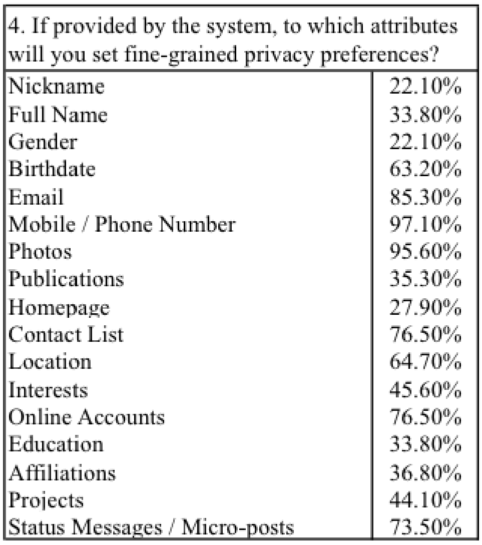


3. If provided by the system, would you set different privacy settings for each part of your profile information? For example: a privacy setting to grant access to your family members to see your personal mobile number and another privacy setting to grant access to your work colleagues to see your email address.	Yes	No
	92.90%	7.10%

Fig. 13. User Study - Question 3

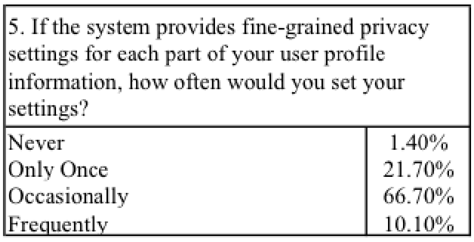
Question 1 (figure 11) shows that 98.60% of the users are aware of privacy settings since they have set them at least once in current Social Web applications. The user who said no and the other user who skipped this question informed us that they are not confident in publishing information in current Social Web applications due to privacy issues, and hence they do not use these type of applications. This illustrates that users are unhappy with current implementations of privacy settings. Question 2 (figure 12) illustrates 88.60% of the users are unhappy to share their profile data with everyone and prefer to grant access to a restricted number of users. Therefore this shows that users require to set privacy settings for their profile information. Question 3 (figure 13) demonstrates that 92.90% require to have fine-grained privacy settings for their personal information which, as mentioned in section 1, current Social Web applications do not provide fine-grained privacy preferences.

In question 4 (figure 14) we asked the users to which parts of their profile information they will most likely set fine-grained preferences. All the attributes contained within the list were chosen revealing that users require to set fine-grained privacy preference for each single information contained in their profile; contact information such as telephone / mobile phone numbers being the most required by 97.10% of the users. 5% of the users provided us with feedback mentioning that they would set different privacy preferences for status messages and micro-posts since they feel confident with publishing micro-posts to a larger audience and they are more concerned to whom they share their sta-



4. If provided by the system, to which attributes will you set fine-grained privacy preferences?	
Nickname	22.10%
Full Name	33.80%
Gender	22.10%
Birthdate	63.20%
Email	85.30%
Mobile / Phone Number	97.10%
Photos	95.60%
Publications	35.30%
Homepage	27.90%
Contact List	76.50%
Location	64.70%
Interests	45.60%
Online Accounts	76.50%
Education	33.80%
Affiliations	36.80%
Projects	44.10%
Status Messages / Micro-posts	73.50%

Fig. 14. User Study - Question 4

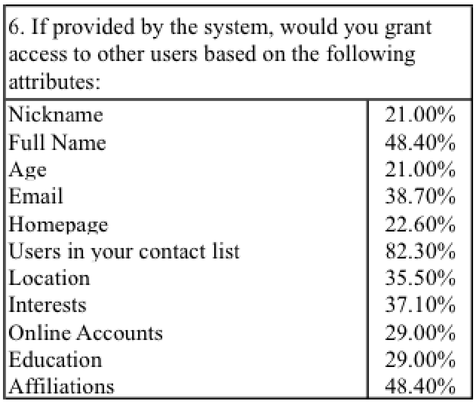


5. If the system provides fine-grained privacy settings for each part of your user profile information, how often would you set your settings?	
Never	1.40%
Only Once	21.70%
Occasionally	66.70%
Frequently	10.10%

Fig. 15. User Study - Question 5

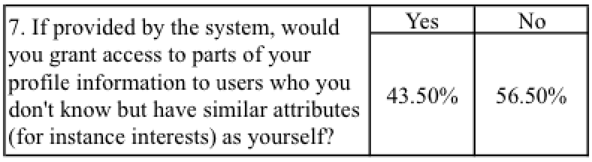
tus messages. This illustrates that users require fine-grained privacy preferences for their status messages. Question 5 (figure 15) demonstrates that 66.70% are willing to set fine grained privacy settings more than once which shows the importance of having a scalable system that provides users to set restrictions to whom they share information with.

In question 6, we asked which attributes users requesting personal information must have in order to share with them private sensitive information. 82.30% of the users answered that they feel confident with sharing information to users in their contact list. Our hypothesis to this result is that users are used to this option since current Social Web applications provide to restrict their information based on contact lists. In order to verify our hypothesis, we omitted to have a contact list in our system but provide users to specify to whom they share information based on similar attributes to theirs. Question 7 inquired whether



6. If provided by the system, would you grant access to other users based on the following attributes:	
Nickname	21.00%
Full Name	48.40%
Age	21.00%
Email	38.70%
Homepage	22.60%
Users in your contact list	82.30%
Location	35.50%
Interests	37.10%
Online Accounts	29.00%
Education	29.00%
Affiliations	48.40%

Fig. 16. User Study - Question 6



7. If provided by the system, would you grant access to parts of your profile information to users who you don't know but have similar attributes (for instance interests) as yourself?	Yes	No
	43.50%	56.50%

Fig. 17. User Study - Question 7

user's prefer to share personal information with users who they don't know but based on similar attributes to theirs, or to users who they already know. Although the results revealed that 56.50% feel more confident in sharing information with people who they know, 43.50% reveal that people are willing to share their information based on similar attributes to people who they don't know. Since the results are almost equal, this also encourages us to develop a system without any contact lists.

6. A Privacy Preference Manager for Creating Faceted Profiles

This section presents *MyPrivacyManager* (screencast online – `http://vmuss13.deri.ie/faceteduserprofiles/screencast/screencast.html`), a Web application that serves as a privacy preference manager for the Social Semantic Web.

6.1. Architecture

MyPrivacyManager was developed 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 that consists of Social Web data formatted in RDF (or any other structured format), we will focus on defining privacy preferences for FOAF profiles. With FOAF profiles, our aim is to illustrate how personal information can be filtered based on privacy preferences to generate faceted profiles.

MyPrivacyManager provides users to manage their privacy preferences and also grants access to user's information when requested. The system therefore restricts evereything by default and grants access to specific information based on the preferences specified by the users.The architecture provides users to:

(1) Authenticate to their *MyPrivacyManager* instance using the WebID protocol and create privacy preferences based on their FOAF profile; and
(2) Authenticate to third party user's *MyPrivacyManager* instance which automatically requests to view the FOAF profile (of the third party) which is filtered based on privacy preferences.

Figure 18 illustrates the *MyPrivacyManager* architecture, which contains:

(1) WebID Authentication: handles user sign-on using the FOAF+SSL protocol (discussed later in this section);
(2) RDF Data Retriever and Parser: retrieves and parses RDF data such as FOAF profiles from WebID URIs;
(3) Creating Privacy Preferences: defines privacy preferences using PPO;
(4) Requesting and Applying Privacy Preferences: 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, hence generating a faceted profile; and
(6) RDF Data store: an ARC2[28] RDF data store to store the privacy preferences[29].

The WebID protocol [50] provides a mechanism whereby users can authenticate using FOAF and SSL certificates. The SSL certificates (which can be self-signed certificates) contain the public key and a URI that points to the location where the FOAF document is stored. Once the user requests to log in *MyPrivacyManager*, the browser prompts the user to select a certificate. The authentication mechanism parses the WebID URI from the certificate and retrieves the FOAF document from its location. The public key in the certificate and the public key in the FOAF file are checked to grant the user access to *MyPrivacyManager* if the public keys match.

MyPrivacyManager uses WebID protocol since it utilises the benefits of URIs where users have a unique identification unlike OpenID[30]. Although OpenID provides a framework where users can log into systems using other system's authentication mechanisms, when users have more than one OpenID account acts as if they identify different persons rather than identifying the same person as how WebID does.

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 or preview his/her faceted profile how it appears to specific users. On the other hand, if the user signed in is a requester, then the faceted FOAF profile of the owner of that particular instance is requested. The *Requesting and Applying Privacy Preferences* 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, hence generating a faceted profile.

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 [5]. Moreover, users can deploy their instances of *MyPrivacyManager* on whichever server they prefer. This approach ensures that the FOAF profile and privacy preferences are private since the user becomes the sole authority of his/her data and nobody can access such data unless he/she is granted access.

[28] ARC2 — `http://arc.semsol.org`

[29] Although ARC2 was used for the implementation of MyPrivacyManager, any RDF store can be used.

[30] OpenID – `http://openid.net/`

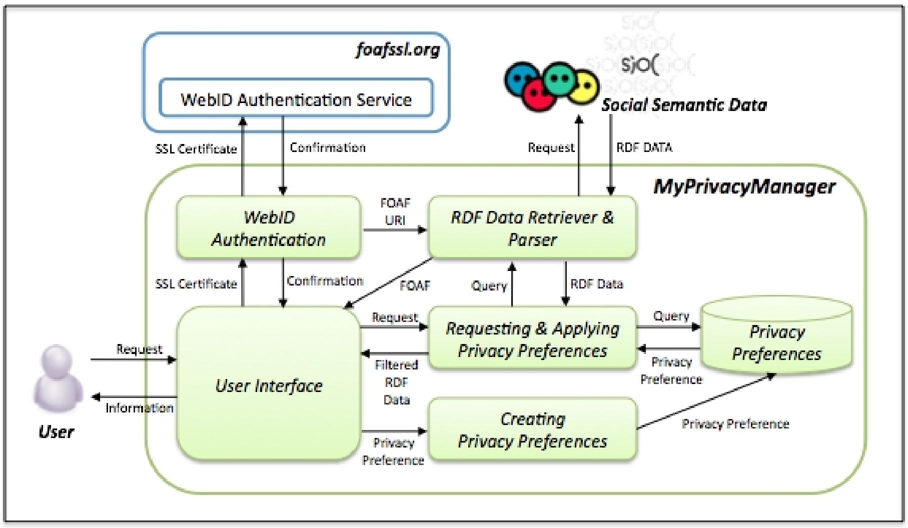


Fig. 18. MyPrivacyManager Architecture

6.2. Implementation

6.2.1. Authentication with the WebID protocol

The WebID protocol implemented in *MyPrivacyManager* uses the libraries provided by foaf.me[31] which calls the WebID authentication mechanism offered by the FOAF+SSL Identity Provider Service[32]. 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.

6.2.2. Creating Privacy Preferences

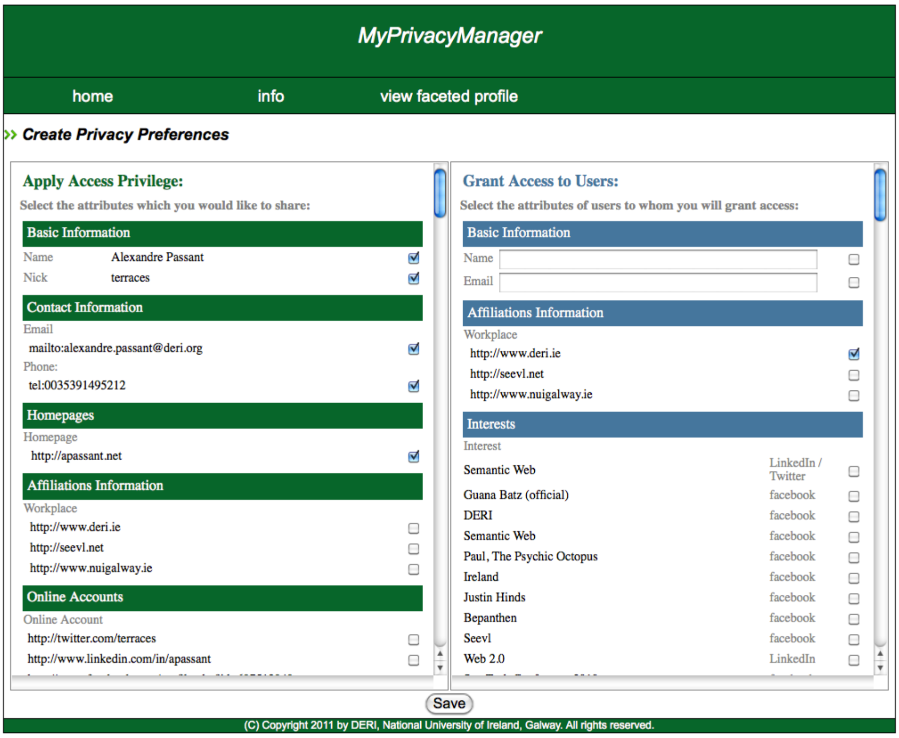
MyPrivacyManager provides users an interface to create privacy preferences for their FOAF profile amalgamated from multiple sources. On loading the interface, the system first retrieves and loads all the vocabularies which are used during the creation of the privacy preferences, mainly the PPO and the WAC vocabularies. Once the vocabularies are loaded, the system retrieves the full FOAF profile (generated from Twitter, LinkedIn and Facebook) from the WebID URI contained within the SSL certificate.The interface then displays (1) the profile attributes 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 (Fig: 19).

The preliminary user study in section 5 clearly shows that users want to specify different parts of their profile information and based on this user study, the system provides profile attributes which the user can share classified as follows:

(1) Basic Information consisting 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.

Moreover, the user study in section 5 demonstrates which attributes the users prefer to specify to whom they want to share their information with. This study shows that users prefer to select specific users from contact list. Since a considerate number of users have selected that they would require sharing information without knowing who the person is, we opted to not

[31] foaf.me — `http://foaf.me/`
[32] foafssl.org — `http://foafssl.org/`



>> *Create Privacy Preferences*

Apply Access Privilege:

Select the attributes which you would like to share:

Basic Information

Name	Alexandre Passant	☑
Nick	terraces	☑

Contact Information

Email
mailto:alexandre.passant@deri.org ☑
Phone:
tel:0035391495212 ☑

Homepages

Homepage
http://apassant.net ☑

Affiliations Information

Workplace
http://www.deri.ie □
http://seevl.net □
http://www.nuigalway.ie □

Online Accounts

Online Account
http://twitter.com/terraces □
http://www.linkedin.com/in/apassant □

Grant Access to Users:

Select the attributes of users to whom you will grant access:

Basic Information

Name ______ □
Email ______ □

Affiliations Information

Workplace
http://www.deri.ie ☑
http://seevl.net □
http://www.nuigalway.ie □

Interests

Interest		
Semantic Web	LinkedIn / Twitter	□
Guana Batz (official)	facebook	□
DERI	facebook	□
Semantic Web	facebook	□
Paul, The Psychic Octopus	facebook	□
Ireland	facebook	□
Justin Hinds	facebook	□
Bepanthen	facebook	□
Seevl	facebook	□
Web 2.0	LinkedIn	□

(C) Copyright 2011 by DERI, National University of Ireland, Galway. All rights reserved.

Fig. 19. The interface for creating privacy preferences in MyPrivacyManager

provide any user contact lists but provide users to specify the attributes of whom they want to share information with. Our aim is to study whether users are satisfied with our approach which provides sharing information to a greater (or less) audience without knowing 'a priori' who the person is and without having the user to maintain user lists. The user evaluation in section 7 shows that users accepted our approach and were satisfied how the system granted access. The attributes, extracted from the FOAF profile, provided by the system which the user can select to whom to share information 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, the clicks on the save button for the system to generate automatically the privacy preference. Hence, the application generates automatically the restrictions, conditions and access space query automatically based on what the user selected. Figure 20 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.

6.2.3. *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 or else to validate their privacy preferences by previewing how other users can view their profile. 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 – either public or private by default.

The sequence in which privacy preferences are requested and enforced is performed as illustrated in figure 21 which consists of:

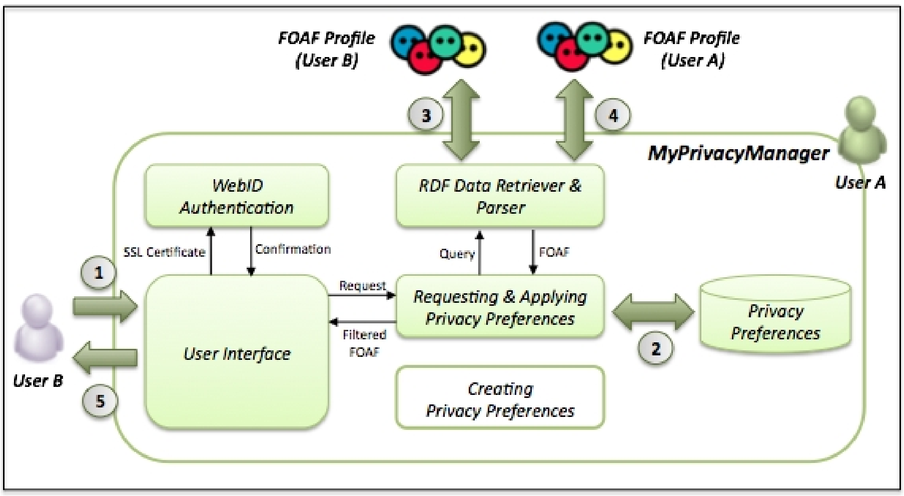


Fig. 21. The sequence of requesting third party FOAF profiles

```
<http://vmuss13.deri.ie/preference#1>
  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. 20. Privacy Preference described using PPO created in *MyPrivacyManager*

(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.

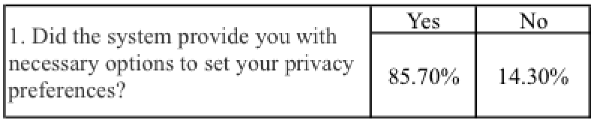
MyPrivacyManager handles each privacy preference separately since each preference may contain different access spaces. Once the system retrieves the privacy preferences, for each preference it tests the access space queries with the requester's FOAF profile. If the access space query on the requester's 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 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. The restrictions and conditions are combined according to what is defined in the privacy preference. For instance if `ppo:appliesToResource` is defined for a resource and in the conditions there is `ppo:resourceAsSubject` defined for the same resource as illustrated in figure 20, then the pattern will only restrict the subject rather than the object as well. Currently, the queries constructed in *MyPrivacyManager* follow the SPARQL 1.0 specification. The pattern formalisation will be modified once the SPARQL 1.1 specification becomes a standard, for instance to include subqueries. This will help in providing more complex queries especially when combining restrictions and conditions together. 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.

7. System Evaluation

The evaluation of our system involved users to create privacy preferences and verifying that what they created corresponds to what other users are allowed to view. The process of the evaluation consisted of a one-to-one interview whereby we commenced by explaining our objectives and overview of our work. We then asked the users to perform 3 tasks which consisted of the following:

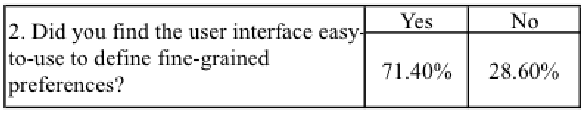
(1) Create 2 or more attributes to users who work at the same workplace as yours;
(2) Create 2 or more attributes to users who are interested in a particular topic; and
(3) Verify how other users view part of your profile based on your privacy preferences.

After the users had completed these tasks, the users were asked to complete an online survey which, together with the results, are illustrated in figures 22 - 26. The users did not have any problems in getting used to the system which in fact it lasted the user between 1 - 2 minutes to complete all the tasks. However, the interviews lasted between 20 to 45 minutes because in each interview each user provided feedback and were also eager to try more privacy preferences than the amount specified in the tasks. Currently only 7 users were interviewed due to limited time and user's availability. However, the user evaluation process is still currently ongoing so that we can gather more feedback and results.



	Yes	No
1. Did the system provide you with necessary options to set your privacy preferences?	85.70%	14.30%

Fig. 22. User Evaluation - Question 1

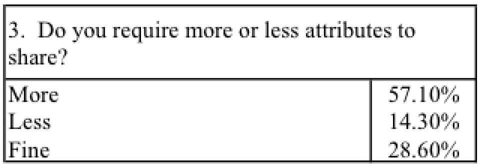


	Yes	No
2. Did you find the user interface easy-to-use to define fine-grained preferences?	71.40%	28.60%

Fig. 23. User Evaluation - Question 2

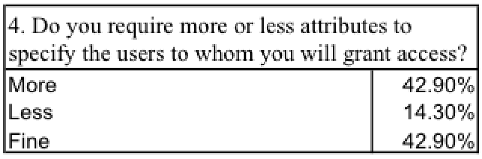
Question 1 (figure 22) asked whether the system provided enough properties to conduct the task of creating privacy preferences and viewing faceted profiles. 85.70% of the users were satisfied with the options, however, 14.30% of the users stated that the interests were irrelevant and preferred to have an option to add new interests. Moreover, they also stated that they would have also preferred to have options to add specific users or user groups. In question 2 (figure 22), 71.40% state that the user interface was user-friendly, however, 28.60% of the users found that the interface provided long lists of interests which required the user having to select many interests. They suggested that interests should be categorised and when a category is select, all the interests in that category are also selected to be shared. Moreover, a user preferred that first they would like to select to whom they want to share first rather than first selecting what they want to share. This requirement is useful to improve the interface by catering for personalisation of user interfaces whereby each user can customise the interface according to their personal preferences.

Question 3 (figure 24) shows that 57.10% of the users require more attributes to share such as photos. This means that the users are eager to use this system to create privacy preferences for more information and not only the ones collected from Twitter, Facebook and LinkedIn. Question 4 (figure 25) demonstrates that 42.90% of the users required more attributes such as location to specify to whom they want to share infor-



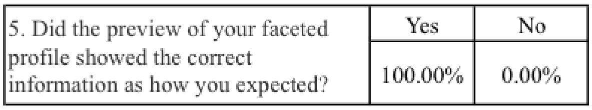
3. Do you require more or less attributes to share?	
More	57.10%
Less	14.30%
Fine	28.60%

Fig. 24. User Evaluation - Question 3



4. Do you require more or less attributes to specify the users to whom you will grant access?	
More	42.90%
Less	14.30%
Fine	42.90%

Fig. 25. User Evaluation - Question 4



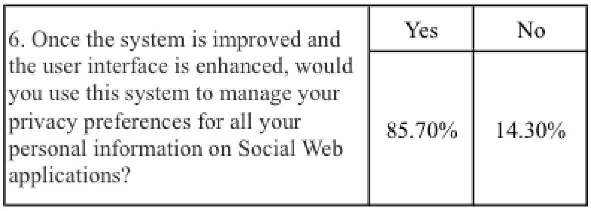
5. Did the preview of your faceted profile showed the correct information as how you expected?	Yes	No
	100.00%	0.00%

Fig. 26. User Evaluation - Question 5

mation. Most of the users suggested to retrieve more interests and not only the ones which they were interested in. Additionally, 42.90% of the users were satisfied with the attributes the system provided.

Question 5 (figure 25) illustrates that all users who were interviewed were satisfied with how the system filtered their profile and how the system generated the faceted profiles for different requesters. This verifies that the system generates the right faceted profile as how the user expected whilst creating their privacy preference.

Question 6 (figure 27) inquired whether the users would use the concept of creating and managing fined-grained privacy preferences for all their personal information on the Social Web. 85.70% answered that they were in favour of creating such fine-grained privacy preferences. This result encourages us to enhance and improve our system to provide as many options as possible for users to be able to create privacy preferences for any data collected and structured from the Social Web. 14.30% will not use this concept due to the tedious task of specifying many privacy preferences for each part of all their information published on the Social Web.



6. Once the system is improved and the user interface is enhanced, would you use this system to manage your privacy preferences for all your personal information on Social Web applications?	Yes	No
	85.70%	14.30%

Fig. 27. User Evaluation - Question 6

8. Conclusion and Future Work

In this paper we described a system providing users full control over their personal user profile allowing them to define and show different *facets* of their profile based on fine-grained privacy preferences. We described the architecture of the user profiling module of the system and the methodology proposed for the aggregation of different user profiles on the Social Web. Moreover, we provided details about the structure of the privacy preference manager - MyPrivacyManager, which allows the specification of the privacy preferences on the profile data. Additionally it also provides users to verify their faceted profiles as visible by other users. The architecture proposed is applicable to any kind of site on the Social Web, and MyPrivacyManager is also platform independent. A user interface for the specification of the privacy preferences has been implemented and evaluated, following the results of a user study that provided us motivation and feedback for this research. A user based evaluation of the system has been conducted and is still ongoing. From the encouraging results of the evaluation, although with a limited number of users, we obtained feedback that help us to improve the user interface and the performance of the system. Apart from the optimisation of the system, as future work in the user interface we would like to include more types of attributes that the users can select. Then we will include a hierarchical structure for the interests organised in categories taken from DBpedia. This would make it easier to navigate and select many interests from a numerous list. As regards the user profiling process we plan to include more information in the user profiles describing the temporal character of the interests and the related concepts or categories.

References

[1] F. Abel, J. De Coi, N. Henze, A. Koesling, D. Krause, and D. Olmedilla. Enabling advanced and context-dependent ac-

cess control in RDF stores. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 1–14. Springer-Verlag, 2007.

[2] F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving Public User Profiles on the Web. In *User Modeling, Adaptation, and Personalization*, pages 16–27. Springer, 2010.

[3] L. Aroyo and G. Houben. User modeling and adaptive Semantic Web. *Semantic Web Journal*, 1(1):105–110, 2010.

[4] M. Assad, D. Carmichael, J. Kay, and B. Kummerfeld. PersonisAD: Distributed, active, scrutable model framework for context-aware services. *Pervasive Computing*, pages 55–72, 2007.

[5] 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.

[6] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, 2008.

[7] S. Berkovsky, T. Kuflik, and F. Ricci. Cross-representation mediation of user models. *User Modeling and User-Adapted Interaction*, 19(1):35–63, Sept. 2009.

[8] D. Berrueta, D. Brickley, S. Decker, S. Fernández, C. Görn, A. Harth, T. Heath, K. Idehen, K. Kjernsmo, A. Miles, A. Passant, A. Polleres, L. Polo, E. U. B. Michael Sintek, and J. G. Breslin. SIOC Core Ontology Specification. W3C Member Submission 12 June 2007, World Wide Web Consortium, 2007. `http://www.w3.org/Submission/sioc-spec/`.

[9] D. Boyn and E. Hargittai. Facebook privacy settings. Who cares? *First Monday*, 15(8), August 2010.

[10] D. Brickley and L. Miller. FOAF Vocabulary Specification 0.91. November 2007.

[11] C. Brooks, M. Winter, J. Greer, and G. McCalla. The massive user modelling system (MUMS). In *Seventh International Conference on Intelligent Tutoring Systems*. Springer, 2004.

[12] P. Brusilovsky and N. Henze. *Open corpus adaptive educational hypermedia*, pages 671–696. Springer, lncs edition, 2007.

[13] P. Brusilovsky, A. Kobsa, and W. Nejdl. *The adaptive web: methods and strategies of web personalization*. Springer-Verlag, lncs edition, 2007.

[14] F. Carmagnola. Handling Semantic Heterogeneity in Interoperable Distributed User Models. *Advances in Ubiquitous User Modelling*, pages 20–36, 2009.

[15] F. Carmagnola, F. Cena, and C. Gena. User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, pages 1–47, Feb. 2011.

[16] F. Carmagnola and V. Dimitrova. An Evidence-Based Approach to Handle Semantic Heterogeneity in Interoperable Distributed User Models. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 73–82. Springer, 2008.

[17] 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.

[18] 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.

[19] P. De Bra, A. Kobsa, and D. Chin. *User Modeling, Adaptation, and Personalization*. Springer, 2010.

[20] R. Denaux, V. Dimitrova, and L. Aroyo. Integrating open user modeling and learning content management for the semantic web. *User Modeling 2005*, pages 9–18, 2005.

[21] J. Fink. *User modeling servers: Requirements, design, and evaluation*. PhD thesis, University of Duisburg-Essen, 2003.

[22] F. L. Gandon and N. M. Sadeh. Semantic Web Technologies to Reconcile Privacy and Context Awareness. In *Proceedings of the 1st French-speaking Conference on Mobility and Ubiquity Computing, UbiMob'04*, 2004.

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

[24] D. Heckmann. Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing. In *ABIS2003, Karlsruhe, Germany*, pages 283–286, 2003.

[25] D. Heckmann, T. Schwartz, B. Brandherm, and A. Kröner. Decentralized user modeling with UserML and GUMO. In P. Dolog and J. Vassileva, editors, *Decentralized, Agent Based and Social Approaches to User Modeling, Workshop DASUM-05 at 9th International Conference on User Modelling, UM2005*, pages 61–66, 2005.

[26] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. Gumo – the general user model ontology. In *User Modeling 2005*, Lecture Notes on Computer Science, pages 428–432. Springer Berlin / Heidelberg, 2005.

[27] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. *Gumo–the general user model ontology*, pages 428–432. Springer, lncs edition, 2005.

[28] B. Heitmann and C. Hayes. Using Linked Data to build open, collaborative recommender systems. In *AAAI Spring Symposium "Linked Data Meets Artificial Intelligence"*, 2010.

[29] 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.

[30] A. Hogan, A. Harth, and A. Pollere. Scalable Authoritative OWL Reasoning for the Web. *International Journal on Semantic Web and Information Systems*, 5(2):49–90, April-June 2009.

[31] 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.

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

[33] A. Kobsa. Generic User Modeling Systems. *The Adaptive Web: Methods and Strategies of Web Personalization*, 4321(LNCS):136–154, 2007.

[34] A. Kobsa and J. Fink. An LDAP-based User Modeling Server and its Evaluation. *User Modeling and User-Adapted Interaction*, 16(2):129–169, July 2006.

[35] A. E. Kobsa. User Modeling and User-Adapted Interaction. The Journal of Personalization Research.

[36] A. Korth and T. Plumbaum. A framework for ubiquitous user modeling. In *IEEE International Conference on Information*

Reuse and Integration, 2007. IRI 2007., pages 291–297. IEEE, 2007.

[37] T. Kuflik. Semantically-enhanced user models mediation: Research agenda. In *Proc. of 5th International Workshop on Ubiquitous User Modeling (UbiqUM'2008), workshop at IUI*, 2008.

[38] B. Mehta and W. Nejdl. Intelligent Distributed User Modelling: from Semantics to Learning. In *UbiDeUM: Proc. of the UM '07 Workshop on Ubiquitous and Decentralized User Modeling*, 2007.

[39] B. Mehta, C. Niederee, A. Stewart, M. Degemmis, P. Lops, and G. Semeraro. Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In A. Ardissono, Liliana and Brna, Paul and Mitrovic, editor, *User Modeling 2005*, Lecture Notes in Computer Science, pages 151–151. Springer Berlin / Heidelberg, 2005.

[40] P. N. Mendes, A. Passant, P. Kapanipathi, and A. P. Sheth. Linked open social signals. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '10, pages 224–231, Washington, DC, USA, 2010. IEEE Computer Society.

[41] P. Nasirifard, V. Peristeras, and S. Decker. Annotation-Based Access Control for Collaborative Information Spaces. *Computers in Human Behavior*, 2010.

[42] S. Nepal, J. Zic, F. Jaccard, and G. Kraehenbuehl. A Tag-Based Data Model for Privacy-Preserving Medical Applications. *Current Trends in Database Technology*, 2006.

[43] C. Niederée, A. Stewart, B. Mehta, and M. Hemmje. A Multi-Dimensional, Unified User Model for Cross-System Personalization. In *Proceedings of the AVI Workshop on Environments for Personalized Information Access*, pages 34–54, 2004.

[44] Oasis. eXtensible Access Control Markup Language (XACML) Version 3.0. 2009.

[45] 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.

[46] 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.

[47] K. Schmidt, L. Stojanovic, N. Stojanovic, and S. Thomas. On enriching ajax with semantics: The web personalization use case. In *The Semantic Web: Research and Applications*, LNCS, pages 686–700. Springer, 2007.

[48] B. Schopman, D. Brickly, L. Aroyo, C. Van Aart, V. Buser, R. Siebes, L. Nixon, L. Miller, V. Malaise, M. Minno, and Others. NoTube: making the Web part of personalised TV. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, pages 1–8, 2010.

[49] 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.

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

[51] M. Szomszor, H. Alani, I. Cantador, K. O'Hara, and N. Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. *The Semantic Web-ISWC 2008*, pages 632–648, 2008.

[52] K. Tao, F. Abel, Q. Gao, and G. Houben. TUMS: Twitter-based User Modeling Service. In *International Workshop on User Profile Data on the Social Semantic Web (UWeb), co-located with Extended Semantic Web Conference (ESWC), Heraklion, Greece*, pages 1–15, 2011.

[53] I. Torre. Adaptive systems in the era of the semantic and social web, a survey. *User Modeling and User-Adapted Interaction*, 19(5):433–486, Nov. 2009.

[54] C. Van Aart, L. Aroyo, Y. Raimond, D. Brickley, G. Schreiber, M. Minno, L. Miller, D. Palmisano, M. Mostarda, R. Siebes, and Others. The NoTube Beancounter: aggregating user data for television programme recommendation. In *Proceedings of the Linked Data on the Web Workshop (LDOW 2009), Madrid, Spain*, pages 1–12, 2009.

[55] J. Vassileva. Distributed user modelling for universal information access. In *Universal access in HCI: Towards an information society for all. Vol. 3. Proceedings*, page 122. Lawrence Erlbaum Associates, 2001.

[56] M. Viviani, N. Bennani, and E. Egyed-Zsigmond. A Survey on User Modeling in Multi-application Environments. In *Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services*, number Section II, pages 111–116. IEEE, Aug. 2010.

[57] B. Zhou, S. Hui, and A. Fong. Web usage mining for semantic web personalization. In *Workshop on Personalization on the Semantic Web (PerSWeb'05)*, pages 66–72, Edinburgh, Scotland, 2005.