Combining Social Networks and Semantic Web Technologies for Personalizing Web Access*

Barbara Carminati, Elena Ferrari, and Andrea Perego

DICOM, Università degli Studi dell'Insubria, Varese, Italy
{barbara.carminati,elena.ferrari,andrea.perego}@uninsubria.it

Abstract. The original purpose of Web metadata was to protect end-users from possible harmful content and to simplify search and retrieval. However they can also be also exploited in more enhanced applications, such as Web access personalization on the basis of end-users' preferences. In order to achieve this, it is however necessary to address several issues. One of the most relevant is how to assess the trustworthiness of Web metadata. In this paper, we discuss how such issue can be addressed through the use of collaborative and Semantic Web technologies. The system we propose is based on a Web-based Social Network, where members are able not only to specify labels, but also to rate existing labels. Both labels and ratings are then used to assess the trustworthiness of resources' descriptions and to enforce Web access personalization.

1 Introduction

The availability of metadata describing Web resources' has been considered as a key issue as soon as the Web became a public information space. Originally, the idea was to use metadata to protect end users from inappropriate and/or harmful content. Released by the W3C in 1996, PICS [1] was the first attempt to define a standard format for such metadata, referred to as *content labels*. Despite PICS has been quickly implemented by MS Internet Explorer and the Netscape browser, resource labeling has not gained success, mainly due to the following reasons. First, resource labeling requires content providers to spend time to describe their resources, and such an effort can be justified only if labels bring real marketing benefits to a content provider. Second, since Web resources' content may frequently change, it is necessary to update content labels accordingly, to be sure that they actually describe the resources they refer to.

However, in recent years the situation has changed, and Web metadata are currently seen by content and service providers as a means to assure the quality of online information. One of the outcomes of such new attitude has been the establishment of the POWDER (Protocol for Web Description Resources) W3C Working Group,[1] aiming at the definition of a new generation of content

* The work reported in this paper is partially funded by the European Community under the QUATRO Plus project (SIP-2006-211001) and by the Italian Ministry of University, Education and Research under the ANONIMO project (PRIN-2007F9437X_004).

[1] Working Group page: http://www.w3.org/2007/powder

E. Bertino and J.B.D. Joshi (Eds.): CollaborateCom 2008, LNICST 10, pp. 126–144, 2009.

labels making use of Semantic Web technologies. Moreover, there currently exist several Web-based Social Networks (WBSNs) providing their members the ability of specifying and sharing metadata (referred to as *tags*), such as, for instance, del.icio.us (`http://del.icio.us`), RawSugar (`http://rawsugar.com`), Flickr (`http://flickr.com`), and Last.fm (`http://last.fm`). Such practice, also known as *social* or *collaborative tagging* [2,3], has the purpose of collecting and sharing opinions about Web resources, and simplifying resource retrieval by organizing resources according to a tag-based browsing criterion.

Although, currently, labels and tags (which we refer to as *Web metadata*) are used, respectively, for quality assurance and tag-based resource classification and browsing, we think that they can have more enhanced applications. In particular, they can be the basis for enhancing access personalization to Web resources. In such a scenario, end users can specify policies determining the actions to be performed by a user agent upon detection of resources associated with given Web metadata. However, in order to achieve this, it is necessary first of all to devise mechanisms able to assess the trustworthiness of Web metadata. We think that collaborative environments and Semantic Web technologies can provide a solution to such issue. In fact, the availability of WBSNs consisting of thousands of users would help not only in increasing the number of labeled/tagged resources, but also in assessing their trustworthiness, based, for instance, on the percentage of labels providing identical descriptions of a resource. Additionally, if WBSN members can express their agreement/disagreement with the descriptions provided by existing labels, this would further help in selecting the most appropriate descriptions for a given resource. By elaborating on these ideas, in this paper we propose a system for collaborative resource labeling and label rating, and we show how it can be exploited for Web access personalization. In our approach, we use the notion of *descriptor* instead of the one of tag, and we denote a *label* as a set of descriptors, modeled according to the POWDER definition. Besides labeling resources, WBSN members can express their dis/agreement about existing labels, by specifying ratings on the contained descriptors. Labels are then statistically analyzed in order to assess the trustworthiness of the contained descriptors. Additional key features of our system are the support for (a) trust policies, making each WBSN member able to denote who he/she considers trustworthy about given topics and/or resource properties, and (b) user preferences, which allow WBSN members to state which actions must be performed on the requested resource, on the basis of the associated descriptors and their trustworthiness. An implementation of our framework is currently carried on in the context of the QUATRO Plus EU project (`http://www.quatro-project.org/`), whose overall goal is to set up an integrated environment for the creation, distribution, and usage of Web metadata.

To the best of our knowledge, currently there does not exist any online service supporting all the features of our proposal. Indeed, most of the existing online communities can be basically considered as recommender systems [4], in that users share resources that they consider relevant, and express personal opinions on them with the purpose of making easier resource retrieval. Examples of these

communities are MovieLens [5] and PHOAKS [6]. In order to simplify the search of shared resources, in the last years several online communities have also provided support for *collaborative tagging*. However, as far as we know, no online community gives its users the possibility of expressing their dis/agreement about existing ratings and tags, with the only exception of MovieLens, where a user can rate (positively or negatively) existing tags. Moreover, none of the abovementioned communities support user preferences for access personalization.

Trust computation is a feature which is supported by some online communities, mainly in order to refine recommendations. Examples are LinkedIn, Orkut, and RepCheck, for personal/professional reputation, FilmTrust, for movie recommendation [7], Moleskiing, for safe skiing [8], and MyWOT (`http://mywot.com`), for Web resource reputation. FilmTrust and Moleskiing are particularly relevant for the scope of our paper. FilmTrust gives its members the possibility of specifying *trust relationships*, denoting how much they trust the opinions on movies of the people they know. Such trust relationships are then used to compute the transitive trust of existing members, in order to weight the relevance of their ratings. Moleskiing adopts a similar approach to personalize recommendations concerning safer skiing. Our trust policies extend the features provided by these online communities, by allowing a user to specify how much he/she trusts another user, in general or with respect to specific descriptors and/or topics. Moreover, trust policies are not only specified in terms of personal relationships existing among WBSN users, but also on user credentials and/or credential attributes.

Finally, user preferences are supported only by MyWOT, an online community for collecting and sharing ratings concerning Web site reputation. MyWOT provides a browser extension allowing subscribed users to associate Web sites with a score concerning the following Web site properties: *trustworthiness*, *vendor reliability*, *privacy*, and *child safety*. Such data are then elaborated by a Bayesan-based algorithm in order to compute an average reputation score. MyWOT allows its members to specify preferences concerning the actions to be performed by the browser, (i.e., *block* or *warning*) upon detection of a Web site with given characteristics. The approach adopted by MyWOT is similar to ours, but some relevant differences are present. First of all, it supports just four descriptors (i.e., trustworthiness, vendor reliability, privacy, and child safety), and it does not give users the possibility of expressing their dis/agreement with claims made by other users. Moreover, differently from our proposal, explicit trust policies or trust relationships are not supported, in that the trustworthiness of rating authors is computed by the system itself.

The remainder of this paper is organized as follows. Section 2 provides a general overview of our approach and describes its architecture. Section 3 introduces our social network model, users' credentials and relationships, and the notions of labels and ratings. Section 4 illustrates trust policies, whereas Section 5 introduces user preferences. Section 6 describes how user preferences are enforced. Finally, Section 7 concludes the paper and outlines future research directions.

2 Overview of the Proposed Approach

In this section, we first introduce the overall framework for a collaborative labelling and rating environment supporting Web access personalization. Then, we discuss the architecture of the proposed system.

2.1 Overall Framework

To support WBSN-based Web access personalization, we propose a framework consisting of five layers (see Figure 1). The Web metadata and ratings layers contain, respectively, resource descriptions and the ratings concerning such descriptions, whereas the users' credentials and relationships layer stores personal information concerning the authors of both Web metadata and ratings. These three layers, as a whole, provide the data used by the upper layers, namely, the trust policies and user preferences layers, which store the rules for determining, respectively, the trustworthiness of Web metadata and the action to be performed by the user agent upon detection of resources associated with Web metadata with given characteristics and trustworthiness.

Each layer in our framework can be seen as a black box, providing a given (set of) service(s) to the upper layers. As such, the only requirement is that layers adopt a standard format for exchanging data, so that they can be implemented by using different technologies. For instance, Web metadata can be encoded by using a variety of formats, but they must be provided to the upper layers by using a standard one. Similarly, as far as the ratings and trust policies layers are concerned, we do not pose any constraint about how reputation and trust are computed, which is totally transparent to the upper layers, which "see" just the results of such computation. The same applies to user preferences.

Such framework has two main advantages. First, our approach can be applied to existing content labeling and collaborative rating systems, by supplying the layers they do not support. For instance, a typical collaborative tagging service supports just the Web metadata layer and, possibly, the one concerning users' credentials and relationships. Ratings, trust policies, and user preferences layers can then be added to enhance its features by exploiting information already stored by the service. The second advantage is that our framework can be used

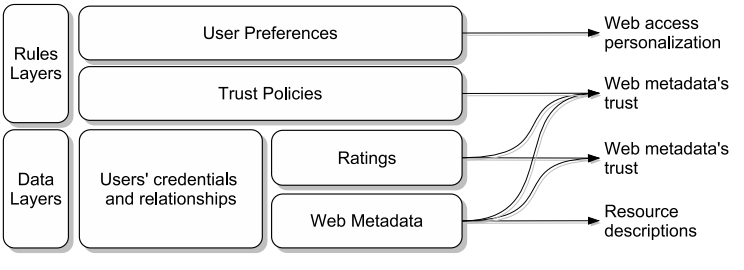

Fig. 1. Layers of the WBSN-based Web access personalization framework

to give end users integrated access to a variety of services. In fact, it may be often the case that an end user is member of several WBSNs and/or online communities, where data are represented by using specific formats, and accessed through specific interfaces. As a consequence, end users can access only separately the resources and services they provide. In contrast, our framework makes it possible to make such systems interoperable by supporting a standard interchange format and a standard set of interfaces, so that end users can transparently access them by using a single tool.

Although there currently exist several frameworks which can be used to enforce the trust policies and user preferences layers of our approach, such as Protune [9] and WIQA [10], in this paper we investigate Semantic Web technologies as the basis of the standard interchange format between our layers. More precisely, RDF/OWL will be used to encode users' credentials and relationships, labels and ratings, whereas trust policies and user preferences will be represented and enforced through N3Logic rules [11].

2.2 System Architecture

The overall architecture of the proposed system is depicted by Figure 2. For simplicity, in the figure, we have omitted the modules in charge of user registration and authentication and those concerning the specification of relationships, labels, ratings, trust policies, and user preferences.

The architecture consists of two main components: the WBSN Management System (WMS) and a WBSN User Agent (WUA). Besides carrying out user registration and authentication, the WMS provides WBSN members the possibility of creating/revoking relationships with other members, specifying *labels* describing the content/characteristics of Web resources, and expressing, by means of *label ratings*, their dis/agreement with existing labels. In contrast, the WUA is in charge of evaluating the trustworthiness of the descriptors contained into the existing labels and notifying such results to WBSN members. Moreover, it allows WBSN members to specify trust policies and user preferences, and it provides an interface to the WMS.

In order to become a WBSN member, a user must register through the registration service provided by the WMS. Then, he/she can provide personal

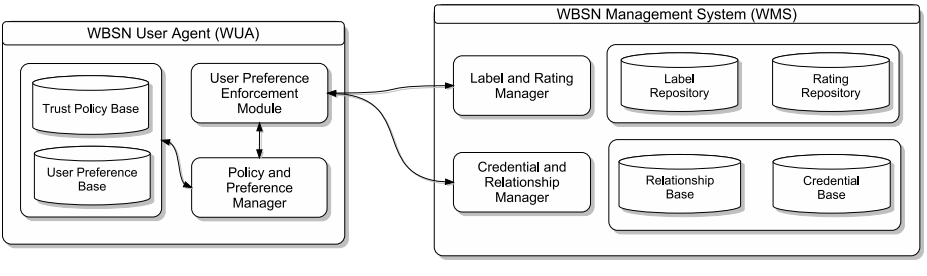

Fig. 2. System architecture

information by importing in the WMS only those of his/her credentials containing information he/she wishes to make publicly available to WBSN members. Users can then create relationships with other WBSN members, specify labels for Web resources, as well as rating existing labels. All this information is stored by the WMS in specific repositories, managed by the *Credential and Relationship Manager* and the *Label and Rating Manager* (see Figure 2).

Resource labels may be of two different types: *owner-defined labels*, that is, labels specified by the owner of the resource(s) they apply to, and *user-defined labels*, that is, labels specified by WBSN members not owning the corresponding resource(s) (see Section 3 for more details). Labels and ratings available in the WBSN are then used to select among those available the most accurate descriptions of the resource content. Such process can be customized by both resource owners and WBSN members through the specification of trust policies (see Section 4).

Finally, WBSN members can specify *user preferences* determining the action to be performed by the WUA when the user tries to access a resource associated with a given set of labels and ratings (e.g., block the access to such resource). User preferences are stored by the WUA in the local *User Preference Base*, which is managed, along with the *Trust Policy Base*, by the *Policy and Preference Manager* (cfr. Section 4). The WUA uses the trust policies in order to determine which descriptors/ratings should be considered when computing descriptors' trust values. In addition, when a WBSN member requests access to a resource, the WUA verifies whether it satisfies the existing user preferences, and performs the corresponding action. These tasks are carried out by the *User Preference Enforcement module* of the WUA (see Figure 2).

3 Users' Credentials and Relationships, Web Metadata and Ratings

In this section, we illustrate the first three layers of our framework (the *data layers*), namely, users' credentials and relationships, Web metadata, and ratings.

3.1 Users' Credentials and Relationships

Typically, in WBSNs a user is associated with two types of information, namely, personal data (such as, first and last name, email address, nationality) and the relationships he/she has with other WBSN members. Therefore, we assume that each WBSN member is denoted by a set of properties (attribute-value pairs), encoded by credentials. Each member of the WBSN may hold one or more credentials. Besides the certified properties, a credential contains the IDs of the Certification Authority (CA) and of the member whom the credential refers to. Finally, the credential is signed by the CA releasing it.

Users' credentials can be modeled by using the FOAF vocabulary [12]. FOAF (Friend of a Friend) is a widely used Semantic Web technology which allows the specification of personal information. In addition, FOAF profiles can be signed

to grant their authenticity, and thus they can be effectively used as certified credentials.

As far as users' relationships are concerned, we model a WBSN $\mathcal{SN}$ as a labeled directed graph, whose nodes denote WBSN members, whereas the labeled edges denote the type of the relationships existing between them. We say that two WBSN members participate in a relationship of a given type rt, if there exists a path connecting them consisting only of edges labeled with relationship type rt. We refer to the length of such path as the *depth* d of the corresponding relationship. If $d = 1$, we say that the relationship is *direct*; if $d > 1$, we say that the relationship is *indirect*. In order to model such notion of relationship, it is possible to use the REL-X ontology[2], which defines an OWL class for the notion of relationship (relx:Relationship), and properties denoting the members (relx:hasMember), type (relx:type), and depth (relx:depth) of a relationship.

An example of WBSN is presented in Figure 3, where nodes correspond to four WBSN members (Alice, Bob, Carol, and David), whereas edges to the relationships existing between them. In the figure, the arrows at both ends of an edge are a shortcut to denote mutual relationships, i.e., the existence of two edges between the same pair of nodes, associated with the same label, and having opposite direction. E.g., the edge connecting A to B denotes the existence of two edges with the same label, one exiting from A and entering in B, and one exiting from B and entering in A.

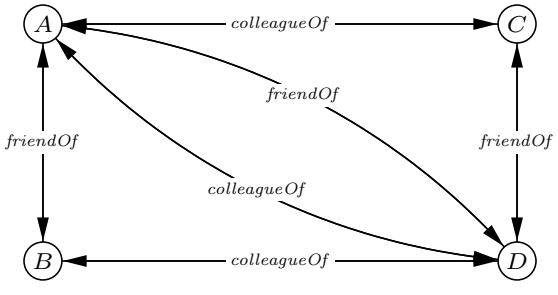

Fig. 3. A small portion of a WBSN

3.2 Web Metadata

As discussed in the previous sections, there exist a variety of Web metadata, adopting different formats and vocabularies. The Web metadata layer in our framework does not put any restriction about the type of supported Web metadata. However, in order to support different Web metadata sources, it is fundamental that they can be transformed into a standard representation. For this purpose, in our framework, we adopt POWDER as an interchange format for Web metadata [13]. POWDER can be used to associate any type of descriptor with a group of resources, and, additionally, to provide meta information about

[2] Namespace URI: `http://www.dicom.uninsubria.it/dawsec/vocs/relx`

such descriptors (such as, who have specified them, when they have been issued, which is their validity period), which can be used as a basis for assessing their trustworhiness. In what follows, we use the traditional term *label* to denote a set of descriptors encoded according to the POWDER format.

Labels describe the content and/or characteristics of a (set of) resources and can be specified by resource owners, or by users belonging to the WBSN. They are identified by a URI and contain a set of *resource descriptors* $rd_1, \ldots, rd_n$, which may be of two different types, namely, *resource property descriptors* and *resource content descriptors*. Resource property descriptors are used to model specific characteristics of the resource (such as the author's name, its title, the language used). They are modelled as pairs $pn = pv$, where pn denotes the name of a resource property and pv denotes the value of pn. In contrast, resource content descriptors are used to denote the relevance of a given topic for describing a resource, and they are expressed as pairs $t = \rho$, where $\rho \in [0, 1]$ denotes the relevance of topic t in describing the considered resource. The set of resources to which a label refers to is denoted by a *URI pattern*, by which it is possible to express statements like "all the resources hosted by `www.example.org`, where the URI path component starts with `foo`".[3] Besides resource descriptors, a label contains the ID of the WBSN member who created it, a timestamp, and, optionally, the validity period for the label.

Example 1. Table 1 presents examples of resource labels, where, for simplicity, the timestamp and validity period have been omitted. Moreover, we denote by LB_n the URI of label n. Labels LB_1 and LB_2 describe all the resources hosted by `www.example.org`. Label LB_1 has been specified by Alice, and it states that she is the author of such resources, that the used language is English, and that topic sport has a relevance equal to 80% in describing their content. Label LB_2 has been specified by Bob, and it states that topic medicine has a relevance equal to 40%. Labels LB_3 and LB_4 describe all the resources having a URI starting with `http://www.example.org/boxing`. Label LB_3, specified by Alice, states that such resources are authored by David, that their title is "Boxing", and that the topic fighting has a relevance equal to 100%, whereas topic violence has a relevance equal to 60%. Finally, label LB_4 is specified by Bob, and it states that such resources are authored by Alice, and topic movies has a relevance equal to 20%, whereas topic violence has a relevance equal to 40%.

Table 1. Examples of labels

URI	Author	URI Pattern	Property Descriptors	Content Descriptors
LB_1	Alice	http://www.example.org*	author = Alice, lang = en	sport = 0.8
LB_2	Bob	http://www.example.org*	$\emptyset$	medicine = 0.4
LB_3	Alice	http://www.example.org/boxing*	author = David, title = Boxing	fighting = 1.0, violence = 0.6
LB_4	Bob	http://www.example.org/boxing*	author = Alice	movies = 0.2, violence = 0.4

[3] URI patterns are specified by using a simplified regular expression syntax, where the wildcard (*) matches a string of $0, \ldots, n$ URI characters.

```
1  @prefix      rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2  @prefix       owl: <http://www.w3.org/2002/07/owl#> .
3  @prefix      wdrs: <http://www.w3.org/2007/05/powder-s#> .
4  @prefix  property: <http://www.example.com/property#> .
5  @prefix      base: <http://mynet.net/labels/lb1#> .
6  @prefix          : <http://mynet.net/members/> .

8  base: a owl:Ontology; wdrs:issuedby :Alice; wdrs:issued "2008-02-12";
       wdrs:validFrom "2008-02-12"; wdrs:validUntil "2009-02-12" .

10 _:Iriset a owl:Class; rdfs:subClassOf [ a owl:Restriction; owl:onProperty
       wdrs:matchesregex; owl:hasValue "http:\/\/www\.example\.org.*" ] .

12 _:D1 a owl:Class; rdfs:subClassOf [ a owl:Restriction; owl:onProperty
       property:author; owl:hasValue :Alice ] .
13 _:D2 a owl:Class; rdfs:subClassOf [ a owl:Restriction; owl:onProperty
       property:lang; owl:hasValue "en" ] .
14 _:D3 a owl:Class; rdfs:subClassOf [ a owl:Restriction; owl:onProperty
       property:sport; owl:hasValue "0.8" ] .

16 _:Iriset rdfs:subClassOf _:D1, _:D2, _:D3 .
```

Fig. 4. OWL-encoding of LB_1 in Table 1

Figure 4 shows the RDF/OWL encoding of LB_1 in Table 1, by using the N3 syntax and according to the POWDER specifications. The ontology header at line 8 encodes the information about who issued the label, when it has been issued, and its validity period. By contrast, the class description at line 10 denotes all the resources having a URI starting with `http://www.example.org`, whereas the class descriptions at line 12-14 denote the resources having Alice as author (line 12), those written in English (line 13), and those where the relevance of topic sport is equal to 80% (line 14). Finally, line 16 states that all the resources having a URI starting with `http://www.example.org` are a subset of those authored by Alice, written in English, and where the relevance of topic sport is equal to 80%. In the RDF/OWL encoding of POWDER, this is how the relationship between a set of resources, denoted by their URIs, and their description is modeled.

3.3 Ratings

According to the proposed interchange format, ratings applying to descriptors in the same label are grouped into a *label rating*. The structure of a label rating is very similar to the one of a label, with the difference that the URI pattern always corresponds to a single resource—i.e., the label being rated—, and no validity period is specified. A rating is considered valid if it has been created after the label it refers to, and such label is not yet expired.

Example 2. Table 2 reports examples of ratings for the labels in Table 1, where the timestamp component of each rating has been omitted for brevity. As in Example 1, we assume that $\text{LB}_1, \ldots, \text{LB}_4$ correspond to the URIs of the labels in Table 1. Rating RT_1 is specified by Bob on label LB_1, and it expresses Bob's agreement about the descriptors stating that Alice is the resource's author (i.e., $(\mathsf{Author} = \texttt{Alice}, 1)$), and that the topic sport has a relevance equal to 80% (i.e., $(\mathsf{sport} = 0.8, 1)$). Also

Table 2. Examples of label ratings

ID	Author	Label's URI	Ratings on Property Descriptors	Ratings on Content Descriptors
RT_1	Bob	LB_1	(author = Alice, 1)	(sport = 0.8, 1)
RT_2	David	LB_1	∅	(sport = 0.8, 1)
RT_3	Carol	LB_2	∅	(medicine = 0.4, 1)
RT_4	David	LB_3	(author = David, 0)	(fighting = 1.0, 1), (violence = 0.6, 0)
RT_5	David	LB_4	(author = Alice, 1)	(violence = 0.4, 1)
RT_6	Carol	LB_4	∅	(movies = 0.2, 0), (violence = 0.4, 1)

rating RT_2 applies to LB_1: it has been specified by David, who agrees that the topic sport has a relevance equal to 80%. Ratings RT_3 and RT_6 have been specified by Carol and they apply to labels LB_2 and LB_4, respectively. In RT_3, Carol agrees that the topic medicine has a relevance equal to 40%. Whereas in RT_6 she disagrees on the fact that the topic movies has a relevance equal to 20%, but she agrees that the topic violence has a relevance equal to 40%. Finally, David specifies also ratings RT_4 and RT_5 concerning labels LB_3 and LB_4, respectively. In RT_4, he disagrees on the fact that he has been claimed to be the author of the labeled resources, and that the relevance of topic violence is equal to 60%, but he agrees that topic fighting has a relevance equal to 100%. In contrast, in RT_5, David agrees that Alice is the author of the labeled resources, and on the fact that topic violence has a relevance equal to 40%.

Figure 5 shows the N3 encoding of RT_1 in Table 2. In order to associate a rating to the statements in label LB_1, they are enclosed into *quoted formulae*,[4] and then the rating (voc:rating)[5] is specified on them. Thus, lines 11-14 specify the rating about the statement according to which Alice is the author of the resource having a URI starting with `http://www.example.org`, whereas lines 15-18 specify the rating about the relevance of topic sport for the same set of resources. Quoted formulae are then used also to specify when the ratings have been issued (line 19), and who issued them (line 20).

As we have discussed in the previous sections, the main purpose of supporting collaborative labeling and rating of Web resources is to identify the most objective descriptions of a resource, to be then used for Web access personalization purposes. This is achieved by aggregating all labels associated with a resource *rsc*, and by computing a *trust value* for each descriptor *rd* contained in the selected labels. Several formulae may be used for trust computation, which might depend also on the possible values used for ratings (e.g., ratings can be binary or scalar, using either discrete or continuous values) and are outside the scope of this paper. Our framework is independent from the adopted one. For that reason, hereafter, we assume the existence of a generic function $\mathcal{T}(rd, rsc)$, which

[4] In N3, quoted formulae are statements delimited by curly brackets, used to represent multiple, and possibly nested, RDF graphs into the same document. This allows one to specify "statements describing other statements", thus providing an alternative to RDF reification.

[5] Here and in the remainder of the paper we use the voc namespace prefix for properties and classes needed to model the notions of our approach.

```
@prefix      rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix       owl: <http://www.w3.org/2002/07/owl#> .
@prefix      foaf: <http://xmlns.com/foaf/0.1/> .
@prefix  dcterms: <http://purl.org/dc/terms/> .
@prefix      wdrs: <http://www.w3.org/2007/05/powder-s#> .
@prefix  property: <http://www.example.com/property#> .
@prefix       voc: <http://mynet.net/voc#> .
@prefix          : <http://mynet.net/members/> .

{
  {
    {[] a owl:Class; rdfs:subClassOf
      [ a owl:Restriction; owl:onProperty wdrs:matchesregex; owl:hasValue
          "http:\/\/www\.example\.org.*" ],
      [ a owl:Restriction; owl:onProperty property:author; owl:hasValue
          :Alice ]
    } voc:rating "1" .
    {[] a owl:Class; rdfs:subClassOf
      [ a owl:Restriction; owl:onProperty wdrs:matchesregex; owl:hasValue
          "http:\/\/www\.example\.org.*" ],
      [ a owl:Restriction; owl:onProperty property:sport; owl:hasValue "0.8"
          ]
    } voc:rating "1" .
  } dcterms:issued "2008-07-11" .
} wdrs:issuedby :Bob .
```

Fig. 5. OWL-encoding of RT_1 in Table 2

takes as input a descriptor *rd* and the corresponding resource *rsc*, and returns it trust value.

4 Trust Policies

Similarly to other collaborative systems, where the collected data are statistically analyzed to assess their reliability, the proposed framework computes the trustworthiness of resources' descriptions based on the associated labels and corresponding ratings. However, in addition to this, our framework makes a user able to explicitly specify which labels/ratings have to be considered during trust computation. In particular, we have identified two different criteria for labels/ratings selection. The first raises by the consideration that a user might prefer to select, for a given resource, only labels/ratings specified by those WBSN members he/she considers trustworthy. For example, for a given Web site, a user might consider trustworthy only his/her direct friends, or only selected colleagues. Moreover, a user might further specify which are the resource topics or properties for which he/she considers trustworthy a given user. For instance, for a given Web site, a user might consider trustworthy with respect to 'sport' only a subset of his/her friends, and, for another topic, say 'music', a different selection of friends.

The second criteria for labels/ratings selection arises by the fact that a resources' owner might wish to suggest those members he/she considers trustworthy with respect to the description/rating of his/her resources. As an example, suppose that a WBSN member owns a Web site dealing with medicine. As resource owner, he/she may wish to allow only medical experts to specify labels for such Web site, and/or rate the associated labels. Providing the resources' owner

with the capability of suggesting trustworthy members brings to the other criteria supported by our framework. Indeed, the second criteria supported by our system is to follow the resource owner suggestions, that is, to select only labels/ratings specified by those members considered trustworthy by the resource owner.

To implement both these criteria of labels/ratings selection, our framework exploits *trust policies*. In particular, it supports two different kinds of trust policies, namely, *user-defined trust policies* and *owner-defined trust policies*. User-defined trust policies implement the first criteria, making thus a user able to specify which are the members he/she judges enough trustworthy, with respect to given topics and/or resource properties, to consider their labels/ratings during the trust computation. In contrast, owner-defined trust policies implement the second criteria, that is, they make a resource owner able to specify which are the members he/she judges enough trustworthy to associate a label/rating with one of his/her resources. How and if these two criteria have to be combined, that is, how user and owner-defined trust policies have to be enforced, depends on the considered scenario. In our approach, to be as flexible as possible, we allow members to specify through their user preferences (cfr. Section 5) whether and how user- and owner-defined trust policies should be combined.

It is interesting to note that, even if user-defined and owner-defined trust policies have different semantics, i.e., the first specify user preferences, whereas the second owner suggestions, syntactically they are similar in that both of them identify a set of members whose labels/ratings have to be considered trustworthy wrt a particular topic/property on a given resource. As such, we make use of a unified syntax to represent both user-defined and owner-defined trust policies. According to this syntax, a user/owner-defined trust policy has to specify the following information: (1) a *URI pattern*, denoting the set of resources to which the policy applies to; (2) *trustworthy members*, that is, a set of WBSN members; (3) a set of *topics* for which the labels/ratings on resources denoted by *URI pattern* and specified by *trustworthy members* are considered trustworthy; (4) a set of *property names* for which the labels/ratings on resources denoted by *URI pattern* and specified by *trustworthy members* are considered trustworthy.

Moreover, the *trustworthy members* can be denoted in three different ways: the first is based on members' IDs, that is, by listing the IDs of those members that have to be considered trustworthy; alternatively, it is possible to exploit relationships existing in the WBSN, that is, to pose constraints on the relationships a user must have in order to be considered trustworthy; finally, another way is by specifying constraints on members credentials. For example, by these three different options, a user is able to (a) state that Ann and Bob are trusted (assuming that names are defined as IDs); (b) identify as trustworthy members those having a "friend of" relationship with Ann of maximum depth 2; or (c) specify as trustworthy members only those whose credentials contain the attribute 'organization' equal to 'University of Insubria'.

Example 3. Table 3 reports examples of trust policies, all applying to the resources hosted by `www.example.org`. TP_1 states that Alice considers trustworthy her direct friends for any topic and for resource property author. In contrast,

Table 3. Examples of trust policies concerning resources having a URI matching pattern `http://www.example.org*` (the URI pattern component has been omitted due to space constraints)

ID	Author	URI Pattern	Trustworthy Members	Topics	Properties
TP_1	Alice	http://www.example.org*	(Alice, *friendOf*, 1)	*	author
TP_2	Bob	http://www.example.org*	Alice	sport	author
TP_3	Carol	http://www.example.org*	expertise = parental_control	violence	*
TP_4	David	http://www.example.org*	(David, *friendOf*, 1), (David, *colleagueOf*, 2)	*	*

TP_2 states that Bob considers Alice trustworthy for topic sport and for resource property author, whereas TP_3 states that Carol considers trustworthy for topic violence and for any resource property only those WBSN members who are experts in parental control. Finally, TP_4 specifies that David considers trustworthy for any topic and resource property only the WBSN members who are, at the same time, David's direct friends, and David's colleagues with a maximum depth equal to 2 (i.e., Alice and Carol, according to Figure 3).

Figure 6 shows the encoding of TP_1 in Table 3 into an N3 rule. More precisely, lines 12-16 correspond to the antecedent of the rule, stating the constraints on the URI pattern (line 13), trustworthy members (line 14), and property/content descriptors (line 15). If such constraints are satisfied, then a label or rating is marked as trustworthy (line 16). Finally, line 17 states that the author of such trust policy is Alice.

Based on such policies, when their authors access a resource *rsc* having a URI matching `http://www.example.org*`, the WUA verifies which descriptors and ratings match the policies (see Table 4). Then, it computes the trust values of the descriptors concerning resource *rsc* by using function $\mathcal{T}(rd, rsc)$ evaluated only on the matching descriptors and ratings.

```
1  @prefix       voc: <http://mynet.net/voc/#> .
2  @prefix      relx: <http://www.dicom.uninsubria.it/dawsec/vocs/relx#> .
3  @prefix       owl: <http://xmlns.com/foaf/0.1/> .
4  @prefix      foaf: <http://xmlns.com/foaf/0.1/> .
5  @prefix       log: <http://www.w3.org/2000/10/swap/log#> .
6  @prefix    string: <http://www.w3.org/2000/10/swap/string#> .
7  @prefix      math: <http://www.w3.org/2000/10/swap/math#> .
8  @prefix      wdrs: <http://www.w3.org/2007/05/powder-s#> .
9  @prefix  property: <http://www.example.com/property#> .
10 @prefix          : <http://mynet.net/members/> .

12 {
13   {
14     ?Resource a foaf:Document; log:uri [ string:startsWith
         "http://www.example.org" ] .
15     ?Relationship a relx:Relationship; relx:hasMember ?Author, :Alice;
         relx:type relx:FriendOf; relx:depth [ math:notGreaterThan "1"] .
16     ?LabelOrRating a [ owl:unionOf ( voc:Label voc:Rating ) ]; wdrs:issuedby
         ?Author; log:includes {[] a owl:Restriction; owl:onProperty
         property:author} .
17   } log:implies {?LabelOrRating voc:isTrustworthy "true"}
18 } wdrs:issuedby :Alice .
```

Fig. 6. N3-encoding of TP_1 in Table 3

Table 4. Labels and ratings in Tables 1 and 2 satisfying (Y) or not satisfying (N) the trust policies in Table 3

	LB_1	LB_2	LB_3	LB_4	RT_1	RT_2	RT_3	RT_4	RT_5	RT_6
Alice	Y	Y	Y	Y	Y	Y	N	Y	Y	N
Bob	Y	Y	Y	Y	Y	N	N	N	N	N
Carol	N	N	N	Y	Y	N	Y	N	N	Y
David	Y	N	Y	N	N	Y	Y	Y	Y	Y

5 User Preferences

Labels' descriptors make end users aware of the content/characteristics of resources. Moreover, the associated trust values, computed based on labels/ratings selected according to user trust policies, make user aware also of descriptor correctness. All these information can then be used by users to decide how a given resource has to be managed, that is, whether it has to be filtered or not. For instance, a user might prefer to block all resources whose labels state that their content is pornographic with trust value at least equal to 80%.

In the proposed framework, this is achieved by means of *user preferences*. These allow a user to specify one or more conditions on resources' descriptors and corresponding trust values, and to state which action has to be performed in case at least one of the specified condition is satisfied. In general, a user preference can be applied to all resources a user is going to access, as well as only to selected resources. Thus, a first component of a user preference is its *scope*, specified as a URI pattern, which forces the system to evaluate the preference whenever a resource denoted by the URI pattern is required. Moreover, user preferences support two types of actions: *block*, which denies the access to resources satisfying at least one of the conditions stated in the user preference; and *notify*, which allows the access, but it forces the system to notify the end user that the resource matches one or more of his/her user preferences.

Regarding the conditions a user can specify, user preferences support constraints on both the property and content descriptors. The former, called *property constraints*, pose conditions on the resource properties and the corresponding trust values. Thus, for instance, it is possible to state that a given resource has to be blocked if the associated descriptors concerning property `author`, and having a trust value greater than 50%, have a value equal to `Alice`. By means of property constraints, a user is also able to specify conditions on the distribution of property descriptors. For instance, a user can enhance the previous preference by specifying that the resource has to be blocked if at least 50% of the `author` descriptor have a trust value greater than 50% and state that Alice is the authors.

Thus, *property constraints* are defined as triples (pc, tc, dc), where: pc is a property constraint of the form $pn \,\mathrm{OP}\, pv$, where pn is a property name, pv is a property value, whereas OP is a comparison operator compatible with pn's domain; tc is a trust constraint of the form $tv \,\mathrm{OP}\, \tau$, where $\tau \in [-1, +1]$ denotes the trust value of the descriptors satisfying pc, and $\mathrm{OP} \in \{=, <, >, \leq, \geq\}$; dc is a

distribution constraint of the form $dv \, \text{OP} \, \delta$, where $\delta \in [0, 1]$ denotes the required percentage of descriptors satisfying pc, and $\text{OP} \in \{=, <, >, \leq, \geq\}$;

In contrast, *content constraints* make a user able to specify conditions on content descriptors and the corresponding trust values. As an example, these constraints allow users to specify that a given resource has to be blocked if the content descriptors with trust value greater than 70% state that topic violence has a relevance greater than 50% in describing the resource.

To support these conditions, *content constraints* consist of two components: a resource content constraint of the form $t \, \text{OP} \, \rho$, where t is a topic, $\rho \in [0, 1]$ denotes the relevance of topic t, $\text{OP} \in \{=, <, >, \leq, \geq\}$, and a trust constraint of the form $tv \, \text{OP} \, \tau$.

Besides specifying constraints on property and content descriptors, user preferences give WBSN members the capability to state how user- and/or owner-defined trust policies have to be taken into account when evaluating descriptors' trustworthiness. More precisely, a user can state whether only user-defined or owner-defined policies, or both/neither of them, must be considered to select the labels and ratings used to evaluate descriptors' trustworthiness. If both user- and owner-defined trust policies must be used, it is also possible to specify whether the descriptors and ratings denoted by user- and owner-defined trust policies must be combined by using union or intersection operator. In addition, we give the end user the possibility of deciding whether all or only some of the owner-defined policies must be taken into account. These preferences are specified by means of the *settings* component, whose syntax is omitted.

Example 4. Table 5 reports examples of user preferences. Preference UP_1, specified by David, requires to block the access to the resources hosted by `www.example.org`, if (a) at least 50% ($dv \geq 0.5$) of the associated descriptors concerning property author, and having a trust value greater than 50% ($tv > 0.5$), have a value equal to `Alice` (author = `Alice`); (b) the content descriptors having a trust value greater than 60% ($tv > 0.6$) state that topics sport and fighting have a relevance greater than 50% (sport > 0.5, fighting > 0.5). Preference UP_1 also states that the descriptors and ratings to be considered when computing descriptors trustworthiness are only those satisfying at least one policy among the owner- and user-defined trust policies (i.e., (`all`, `all`, $\cup$)). Differently from UP_1, preference UP_2, specified by Alice, does not include content constraints, and it states that, when evaluating descriptors' trustworthiness, the WUA must select only the descriptors and ratings satisfying at least one policy among (a) the user-defined trust policies or (b) those owner-defined policies selected at run-time by Alice (this is denoted by (`all`, `some`, $\cup$)). Preference UP_3, specified by Bob, includes both property and content constraints, and it states that, when evaluating descriptors trustworthiness, only the descriptors and ratings satisfying at least (a) one among the owner-defined trust policies and (b) one among the user-defined trust policies must be considered (i.e., (`all`, `all`, $\cap$)). Finally, preference UP_4, specified by Carol, includes just content constraints, and it asks the WUA to perform a 'block' action in case it is satisfied. For evaluating

Table 5. Examples of user preferences

ID	Author	Scope	Property Constraints	Content Constraints	Settings	Action
UP_1	David	http://www.example.org*	(author = Alice, $tv > 0.5$, $dv \geq 0.5$)	(sport > 0.5, $tv > 0.6$), (fighting > 0.5, $tv > 0.6$)	(all, all, $\cup$)	block
UP_2	Alice	http://www.example.org*	(author = Bob, $tv = 1.0$, $dv = 1.0$)	$*$	(all, some, $\cup$)	notify
UP_3	Bob	http://www.example.org*	(author $\neq$ Alice, $tv > 0.5$, $dv > 0.5$)	(sport > 0.4, $tv > 0.8$), (medicine > 0.8, $tv > 0.8$)	(all, all, $\cap$)	notify
UP_4	Carol	http://www.example.org*	$*$	(violence > 0.8, $tv > 0.2$)	(all, none, $\cup$)	block

descriptors' trustworthiness, UP_4 states that only user-defined trust policies must be considered (i.e., (all, none, $\cup$)).

Figure 6 shows the encoding of UP_1 in Table 5 into an N3 rule. More precisely, line 12 denotes the trust policies to be considered (in this case both owner- and user-defined, combined by using OR). If, after having evaluated the rules corresponding to the selected trust policies, their conclusions (i.e., the statements inferred from the rules) satisfy the constraints on property/content descriptors in the user preference (lines 13-18), then the WUA is asked to perform the "block" action (line 19). Finally, line 20 states that the author of such user preference is David.

6 User Preference Enforcement

In general, when a member requests access to a resource, the WUA verifies whether the requested resource satisfies one or more of his/her user preferences, if any. If this is the case, WUA performs the action(s) specified in the satisfied user preference(s). In particular, if the satisfied user preferences specify different actions, we assume that the 'block' action prevails over the 'notify' one. In case no user preferences are satisfied, the WUA performs the *default action*, which can be either `block` or `notify`, and it is set by the end user in the WUA's configuration parameters.

User preference enforcement starts by retrieving the set of user preferences specified by the member requesting the resource. Among these user preferences, the WUA considers only those that apply to the requested resource, that is, those whose scope includes the requested resource. In case there do not exist any user preferences applying to the requested resource, the WUA authorizes the access. Otherwise, user preferences are evaluated to determine the set of actions ACTs to be performed. In case ACTs is empty, then the default action is performed. In case it contains at least a block action, the resource is blocked; otherwise the WUA performs the notify action.

The main steps carried out to evaluate each user preference are depicted in Figure 8.

```
@prefix       voc: <http://mynet.net/voc/#> .
@prefix       owl: <http://xmlns.com/foaf/0.1/> .
@prefix      foaf: <http://xmlns.com/foaf/0.1/> .
@prefix       log: <http://www.w3.org/2000/10/swap/log#> .
@prefix    string: <http://www.w3.org/2000/10/swap/string#> .
@prefix      math: <http://www.w3.org/2000/10/swap/math#> .
@prefix      wdrs: <http://www.w3.org/2007/05/powder-s#> .
@prefix  property: <http://www.example.com/property#> .
@prefix         : <http://mynet.net/members/> .

{
  { ?TrustPolicy a [ owl:unionOf (voc:UserDefinedTrustPolicy
      voc:OwnerDefinedTrustPolicy) ];
    log:supports {
      ?Resource a foaf:Document; log:uri [ string:startsWith
          "http://www.example.org"] .
      { ?Resource a foaf:Document; property:author :Alice . } voc:trustLevel
          [ math:greaterThan "0.5"]; voc:distribution [ math:notLessThan
          "0.5"] .
      { ?Resource a foaf:Document; property:sport [ math:greaterThan "0.5"]
          . } voc:trustLevel [ math:greaterThan "0.6"] .
      { ?Resource a foaf:Document; property:fighting [ math:greaterThan
          "0.5"] } voc:trustLevel [ math:greaterThan "0.6"] .
    } .
  } log:implies { voc:Wua voc:action voc:Block } .
} wdrs:issuedby :David .
```

Fig. 7. N3-encoding of UP_1 in Table 5

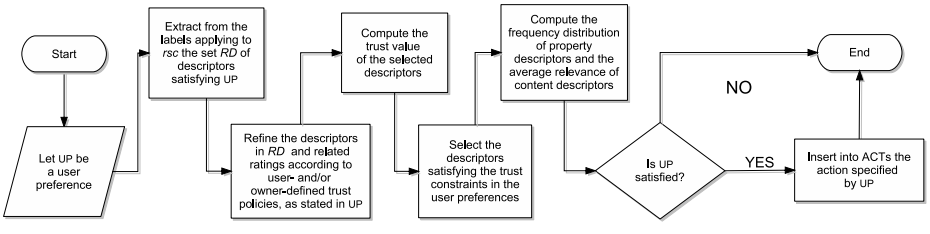

Fig. 8. Main steps of user preference enforcement

In particular, given a user preference UP, the first step retrieves those property and content descriptors, that are relevant for the property and content constraints specified in UP.

For instance, consider user preference UP_1 in Table 5, and suppose that David requests access to a resource *rsc* having URI `http://www.example.org/boxing/`, owned by Alice. In order to determine the action to be performed on *rsc*, the WUA first retrieves the associated labels, and then extracts from them the set of descriptors concerning property author or topics sport and fighting (i.e., some descriptors of labels LB_1, LB_3, and LB_4 in Table 1).

In the second step, the retrieved property and content descriptors are refined according to the owner and user-defined trust policies, combined together according the setting specified in UP. The enforcement also retrieve ratings of refined descriptors by enforcing again the owner and user-defined trust policies.

As example, according to the settings in UP_1, WUA selects only the descriptors and associated ratings satifying one among David's trust policies (i.e., TP_4) or

the owner-defined trust policies (i.e., TP_1)—see Table 3. In this case, all the descriptors selected in the previous phase satisfy TP_1 or TP_4, whereas the selected ratings are RT_1, RT_2, RT_4, and RT_5 (see Table 2).

Once gathered all ratings satisfying the trust policies, the WUA computes the trust values of those descriptors resulting from second step (see third step in Figure 8). Then, it further refines the descriptors by removing those that do not satisfying the trust constrains specified in the UP. Note that both the property and content constraints pose conditions on trust value, thus this refinement is performed on both the property and content descriptors. To determine whether UP is satisfied or not, it is necessary to compute also the average relevance of the content descriptors as well as the frequency distribution of the property descriptors. These computations are performed in the fifth step in Figure 8.

Thus, referring to our example, the WUA computes (a) the trust value $tv_{rd,rsc}$ of each selected descriptor *rd* for resource *rsc*, (b) the average relevance $\overline{\rho}_{cc}$ of the topic in each selected content descriptor *cc*, and (c) the percentage $\delta_{\mathsf{author=Alice}}$ of the set of property descriptors having a trust value greater than 50% and satisfying $\mathsf{author} = \mathtt{Alice}$. For the purposes of our example, we assume the following values: $tv_{rd,rsc} = +1$, if *rd* corresponds to $\mathsf{sport} = 0.8$, $\mathsf{fighting} = 1.0$, or $\mathsf{author} = \mathtt{Alice}$; $tv_{rd,rsc} = -1$, if *rd* corresponds to $\mathsf{author} = \mathtt{David}$; $\overline{\rho}_{cc} = 0.8$, if *cc* concerns topic sport; $\overline{\rho}_{cc} = 1.0$, if *cc* concerns topic fighting; $\delta_{\mathsf{author=Alice}} = 1.0$.

Finally, the WUA verifies if all the property and content constraints are satisfied. If this is the case, the corresponding action is inserted into ACTs, and the process ends. Otherwise, the process ends without inserting a new action into ACTs.

In our example, UP_1 is satisfied, the WUA blocks the access to resource *rsc*.

7 Conclusions and Future Work

In this paper, we have presented a WBSN environment supporting collaborative labeling and rating, where the labels/ratings specified by its members are used to compute the trust value of resources' descriptors and to enforce Web access personalization. Key features of our system are the support for (a) trust policies, making each WBSN member able to denote who he/she considers trustworthy about given topics and resource properties, and (b) user preferences, which allow WBSN members to determine which action must be performed by the user agent on the requested resource, upon detection of descriptors with given characteristics and a given trust value.

An implementation of our framework is currently carried on in the context of the QUATRO Plus EU project (`http://www.quatro-project.org/`), whose overall goal is to set up an integrated environment for the creation, distribution, and usage of Web metadata.

In order to improve the accuracy of trust computation, an issue we plan to address concerns how the *specificity* of a label should affect the trustworthiness of the contained descriptors. For instance, labels $\text{LB}_1, \ldots, \text{LB}_4$ in Table 1 all apply to the same resource *rsc*, having URI `http://www.example.org/boxing/`.

More precisely, such labels apply to two sets of resources, denoted, the former, by URI pattern `http://www.example.org*` (LB_1 and LB_2), and the latter by `http://www.example.org/boxing*` (LB_3 and LB_4). Since the latter is included in the former, we say that it is more *specific* with respect to *rsc*'s URI. In such a case, it should be reasonable that the descriptors in LB_3 and LB_4 are considered more trustworthy than those in LB_1 and LB_2. Note that such specificity principle may be also applied to user preferences, in order to determine which action should be performed by the user agent on the requested resource. A deep study of these issues will be part of our future work.

References

1. Resnick, P., Miller, J.: PICS: Internet access controls without censorship. Commun. ACM 39(10), 87–93 (1996)
2. Golder, S.A., Huberman, B.A.: The structure of collaborative tagging systems. The Computing Research Repository (CoRR) abs/cs/0508082 (2005)
3. Voß, J.: Tagging, folksonomy & Co – Renaissance of manual indexing? The Computing Research Repository (CoRR) abs/cs/0701072 (2007)
4. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering 17(6), 734–749 (2005)
5. Sen, S., Lam, S.K., Rashid, A.M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F.M., Riedl, J.: Tagging, communities, vocabulary, evolution. In: CSCW 2006, pp. 181–190 (2006)
6. Terveen, L., Hill, W., Amento, B., McDonald, D., Creter, J.: PHOAKS: A system for sharing recommendations. Commun. ACM 40(3), 59–62 (1997)
7. Golbeck, J.A.: Generating predictive movie recommendations from trust in social networks. In: iTrust 2006, pp. 93–104 (2006)
8. Avesani, P., Massa, P., Tiella, R.: A trust-enhanced recommender system application: Moleskiing. In: 2005 ACM Symposium on Applied Computing (SAC 2005), pp. 1589–1593 (2005)
9. Bonatti, P.A., Olmedilla, D.: Driving and monitoring provisional trust negotiation with metapolicies. In: POLICY 2005, pp. 14–23. IEEE CS, Los Alamitos (2005)
10. Bizer, C., Cyganiak, R., Maresch, O., Gauss, T.: The WIQA – Web Information Quality Assessment framework. Technical report, Freie Universität Berlin (2006), `http://www4.wiwiss.fu-berlin.de/bizer/WIQA/`
11. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A logical framework for the World Wide Web. Theory and Practice of Logic Programming 8(3), 249–269 (2008)
12. Brickley, D., Miller, L.: FOAF vocabulary specification 0.91. RDF Vocabulary Specification (November 2007), `http://xmlns.com/foaf/0.1`
13. Archer, P., Smith, K., Perego, A.: Protocol for Web description resources (POWDER): Description resources. W3C Working Draft, World Wide Web Consortium (October 2008), `http://www.w3.org/TR/powder-dr`