The Quality Social Network: A Collaborative Environment for Personalizing Web Access

Andrea Perego Barbara Carminati Elena Ferrari
Dipartimento di Informatica e Comunicazione
Università degli Studi dell'Insubria, Varese, Italy
Email: firstname.lastname@uninsubria.it

***Abstract*—In this paper, we present a collaborative social networking environment, referred to as *Quality Social Network* (QSN), which enhances the social tagging paradigm by using it as a basis to evaluate the quality of Web resources, on the basis of the user preferences specified by each QSN member. Such features give end users the ability of being aware of the "quality" of the resources they are accessing, based on the opinions of the members of their community, and of being informed whether such resources can be safely used, according to the requirements specified by end users themselves.**

Besides illustrating the main characteristics of the QSN and its architecture, we describe its prototype implementation, carried out in the framework of the QUATRO Plus EU project, and its application to a use case scenario, involving groups of teenagers from three different European countries, acting as Youth Panels of the Safer Internet EU Programme.

I. Introduction

The widespread diffusion and success in the last few years of collaborative, user-centric services, such has blogs and social networks, has been hailed by many as a great step forward in the evolution of the Web. The services and applications of the so-called Web 2.0, by enforcing the *Web as a platform* paradigm, allow any end user, independently from his/her technical skills, to become a Web content producer, and have transformed the Web from an information to a social space. Despite its unquestionable advantages, this has made the Web a space which is even unsafer for end users with respect to the past. In fact, although search engines have become more and more effective in exploring the Web, end users must still determine the "quality" of the accessed resources by themselves, even when they do not have the required expertise. Consequently, when the Web is used for sensitive purposes (e.g., finding medical information) or by inexpert or naïve users (e.g., minors), this may lead to harmful results. For these reasons, the Web as a whole is still considered by many as a source of unreliable and untrustworthy information, thus preventing the exploitation of its full potentialities.

The QUATRO Plus EU project (http://www.quatro-project.org) tries to address these issues. The goal of the project is to set up a software platform which exploits Semantic Web and Web 2.0 technologies to make end users aware of the quality of the resources they access. More precisely, the QUATRO Plus plaform provides services for the creation, retrieval, and authentication of metadata describing Web resources, referred to as *labels*, as well as front-end tools in charge of displaying labels to end users, and to evaluate them according to user-defined criteria.

In this paper, we illustrate one of the main components of the QUATRO Plus platform, the *Quality Social Network* (QSN), a collaborative environment which enhances the social tagging paradigm [1], [2] by using it as a basis to evaluate the quality of Web resources. More precisely, QSN members are given the ability of associating labels with Web resources, and to rate labels specified by other users or third parties. The collected information is then statistically analyzed in order to assess labels' trustworthiness. Finally, QSN members can specify *user preferences*, determining the action to be performed by a user agent upon detection of resources associated with given labels and with given trust values.

It is worth noting that the exploitation of resource descriptions in order to enforce Web access personalization is a feature that is not supported by existing social tagging services, which have the main purpose of allowing their member to organize and browse resources according to a content-based criterion. To the best of our knowledge, the only exception is MyWOT (http://www.mywot.com), a service which gives its subscribers the ability to rate resources with respect to four criteria: trustworthiness, vendor reliability, privacy, and child safety. MyWOT subscribers can then specify preferences determining whether the browser should block access to a given resource, or should simply return a warning message. Despite the existence of some similarities, the approach adopted by MyWOT is quite different from ours. In particular, the QSN does not put any constraint on the vocabularies to be used and it supports user preferences which are far more flexible than the ones of MyWOT, thus making end users able to precisely express the quality requirements to be satisfied.

The QSN is an application, adapted to the requirements of a specific domain, of the general framework for Web access personalization we have proposed in an earlier paper [3]. Differently from [3], which aimed at defining the layers and services supported by the framework, in this paper we focus on the description of the prototype implementation of the QSN, providing also a use case scenario (see Section VII).

The remainder of this paper is organized as follows. Section II provides a brief description of the QUATRO Plus platform, whereas Section III illustrates the main characteristics

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8368
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8368

of the QSN and its architecture. Section IV is devoted to QSN labels, ratings, and their evaluation. User preferences' syntax and semantics is illustrated in Section V, whereas Section VI deals with user preference enforcement. Section VII illustrates the QSN prototype implementation and the use case scenario on which it is currently being tested. Finally, Section VIII concludes the paper and outlines future research directions.

II. The QUATRO Plus Platform

The purpose of the QUATRO Plus project is to set up software platform in charge of the production and discovery of metadata about online resources, referred to as *labels*, which are then delivered to end users in order to make them aware of the content and characteristics of the resources they access.

In QUATRO Plus, labels are represented by using the standard format defined by the *Protocol for Web Description Resources* (POWDER) W3C Working Group, whose specification has reached the status of W3C Recommendation in September 2009. The purpose of *POWDER documents* is to allow the specification of machine-understandable descriptions of a *set* of resources, denoted by putting constraints on their IRIs.[1] POWDER documents has been designed in order to be as flexible as possible with respect to how a resource description is specified. More precisely, they can support both terms defined in RDF vocabularies and user-defined tags. Finally, POWDER documents includes provenance information, which can be used both to authenticate a POWDER document, and to determine its trustworthiness.

The QUATRO Plus consortium has played an active role in the POWDER WG, not only by contributing in defining the POWDER specification, but also by developing a set of software tools able to process POWDER documents.[2] For a more detailed description of the POWDER technology, which is out of the scope of this paper, we refer the reader to the POWDER specification [8]–[10].

Following, we provide just a brief overview of the QUATRO Plus platform, in order to contextualize the QSN. For a more detailed description, we refer the reader to [11].

The architecture of the QUATRO Plus platform, depicted by Figure 1 consists of a set of distributed tools and services, communicating through SOAP interfaces. The two core components are the QUATRO Plus proxy (QUAPRO+) and the QSN. QUAPRO+ is in charge of detecting labels specified by third parties, referred to as *labeling authorities*, and to authenticate them by contacting the Data Access Interface (DACC) of the corresponding label repositories. By contrast, the QSN, which we illustrate in this paper, has the primary purpose of supplying to the QUATRO Plus platform the benefits of collaborative labeling and rating by integrating social networking features.

[1]In this paper, we use the notion of IRI (*Internationalized Resource Identifier*) [4] as a generalization of the more common one of URI [5], and its subclasses, i.e., URLs [6] and URNs [7].

[2]Such tools are available from the WG Web site: http://www.w3.org/2007/powder.

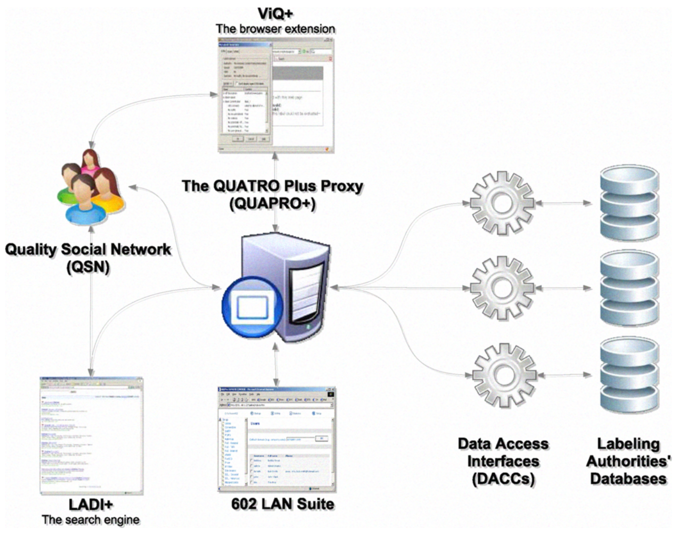

ViQ+
The browser extension
The QUATRO Plus Proxy (QUAPRO+)
Quality Social Network (QSN)
Data Access Interfaces (DACCs)
Labeling Authorities' Databases
LADI+
The search engine
602 LAN Suite

Fig. 1. The architecture of the QUATRO Plus platform

The QUATRO Plus platform includes also a set of front-end tools, accessing the services provided by QUAPRO+ and the QSN. Such tools are:

- ViQ+, a browser extension built on top of the Greasemonkey script engine (http://www.greasespot.net), which is in charge of displaying to end users the labels associated with Web resources and to enforce user preferences on the client side;
- LADI+, a search engine wrapper which annotates search results with the metadata returned by QUAPRO+ and the QSN;
- the LAN 602 Suite, a set of intranet tools which makes use of the data returned by QUAPRO+ and the QSN to enhance the supported content and mail filtering mechanisms.

III. The Quality Social Network

The QSN is a collaborative environment aiming at enforcing Web access personalization based on metadata describing the content and/or characteristics of Web resources, referred to as *labels*, and their trustworthiness. For this purpose, the QSN provides end users the possibility not only of associating labels with Web resources, but also to express their dis/agreement about existing labels by assigning a *rating* to them. The collected labels and ratings are then used as a basis for computing the trustworthiness of existing labels, and for returning, to either end users or any online service submitting a request, the *aggregate view* of labels associated with a given resource. The aggregate view of a set of labels consists of the set of *descriptors* specified in such labels, associated with a trust value, computed based on the set of parameters which are decided by system administrators (see Section IV for more details). Such feature aims at addressing a primary requirement for the use of Web metadata to enhance the access to the Web. In particular, end users will be given the possibility of specifying *user preferences* stating which action should

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8368
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8368

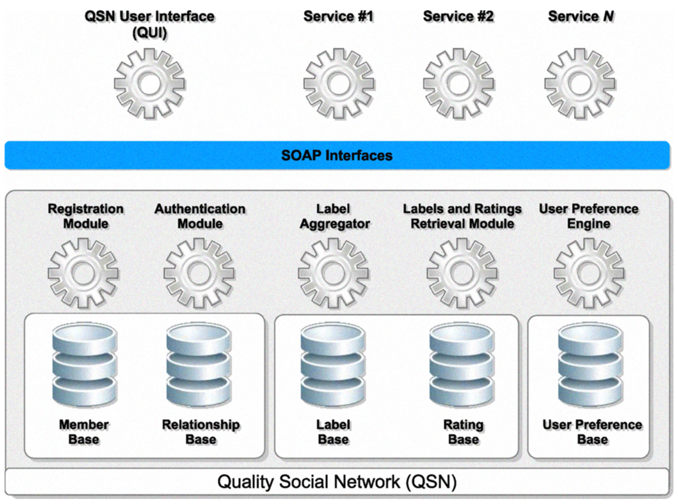


Fig. 2. QSN architecture

be performed by a user agent upon detection of resources associated with given labels having a given trust value.

In addition, the QSN supports the basic features of Web-based social networks, that is, the possibility of specifying and sharing personal data and of establishing relationships with other QSN members. Personal data are encoded as FOAF profiles [12], where the supported relationship types are represented by using terms from the RELATIONSHIP vocabulary [13].

Although a public version of the QSN will be available for any Web user, similarly to "traditional" Web-based social networks, the QSN has been designed as a social networking service which can be run on the network of institutions and/or organizations. For this reason, the QSN supports a set of configuration options which can be chosen by system administrators in order to customize local installations of the QSN with respect to the requirements of the institution and/or organization running it. For example, system administrators can choose the services to enabled and disabled, they can decide whether registration is open to any Web user or only upon invitation made by QSN members, they can select the trust algorithm to be used to compute labels' trustworthiness.

The QSN architecture, depicted by Figure 2 consists of three main components:

1) A set of bases, in charge of storing system data. More precisely, they store QSN members' personal data, relationships, labels, ratings, and user preferences.
2) A set of modules, providing the core QSN services, namely, user registration and authentication, labels and ratings retrieval, label aggregation, and user preference evaluation and enforcement.
3) A set of SOAP interfaces, providing access to the QSN services and data from external Web services, described by a WSDL file.

It is important to note that, by adopting the Web service paradigm, the actual QSN architecture is totally transparent to the software agents making use of QSN services. This means that QSN services can be integrated into other systems, independently from their architecture and on whether they are implemented with technologies different from the ones used in the QSN. The only requirement for such system is to set up a SOAP client able to communicate with the relevant QSN service(s). Based on the same principles, the QSN User Interface (QUI) is built as an external component which makes use of QSN services via the provided SOAP interfaces. This means that there is no requirement about the machine hosting the QUI, which can be run by a remote server, different from the one hosting the QSN. For the same reason, it is possible to have multiple QUIs, run by different machines, all accessing the same QSN service.

Another relevant characteristic of the QUI is its independence from the actual markup language use to present it. More precisely, the structure of the user interface (i.e., its sections and the components of each section) are defined by using a set of XML elements and attributes defined for this purpose. Then, specific XSL transformation rules [14] are used to convert it into the final interface format. Such a strategy can then be used to generate interfaces in any XML-based language, such as XHTML and XUL.[3] The use of XML-based technologies for the QUI allows us also to simplify the support for localized versions of the interface. For this purpose, we adopt the approach used to localize Mozilla and Firefox extensions, based on XML *entity references*.[4] More precisely, the textual content of the QUI (labels, captions, tooltips, etc.) can be substituted by conventional codes (i.e., XML entity references), defined in separate ENT files, along with their textual correspondence. The conversion from a given XML entity reference and the corresponding text is automatically performed as soon as the XML-based QUI is converted into the final format. Thanks to this, it is possible to build the same version of the QSN for a different language simply by providing an additional ENT file, where the codes are associated with the translation of the corresponding textual content. The proper ENT file will then be automatically loaded based on the current localization of the end user's browser.

The diagram in Figure 3 shows the relationships among the XML-based QUI, the ENT files containing localized text of the QUI, the XSLT files containing transformation rules, and the user agents through which the QUI might be accessed (in the figure, a Web browser and a hypothetical Mozilla / Firefox extension).

IV. Labels, Ratings, and Trust

As any of the tools and services of the QUATRO Plus platform, the QSN represents labels according to the format defined for POWDER documents. More precisely, a QSN label consists of the following main components:

[3]XUL (XML User Interface Language) is the markup language used in Mozilla and Firefox to describe the user interface as well as the interface of Mozilla and Firefox extensions. For more details, we refer the reader to [15].

[4]For this notion, we refer the reader to the W3C Recommendation titled "Extensible Markup Language (XML) 1.0" [16].

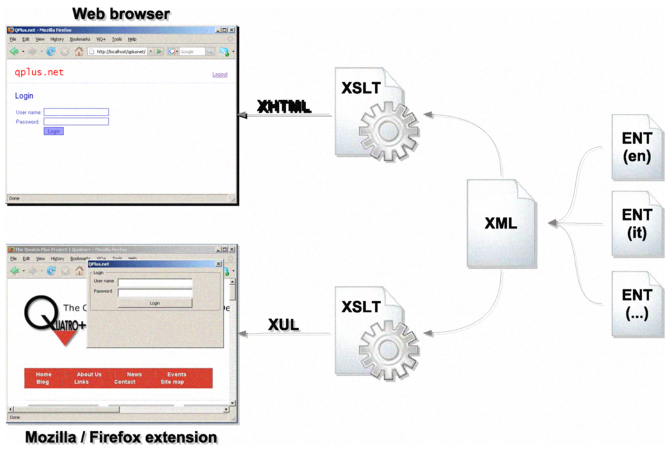


Fig. 3. Transformation and localization process of the XML-based QUI

- *Author*: the QSN member who specified the label, denoted by the IRI of his/her QSN FOAF profile;
- *IRI pattern*: a pattern denoting a set of resources in terms of their IRIs
- *Description*: a set of descriptors, denoting the characteristics/content of the resources denoted by IRI pattern.
- *Summary*: a textual message, to be returned to the end user, which provides a human-readable description of the label's meaning.
- *Issue date*: the timestamp of the instant when the label has been specified.
- *Validity period*: an optional component denoting since and until what date the label must be considered as valid.

Figure 4 provides an example of QSN label, represented as a POWDER document. Its semantics can be informally expressed as follows:

> On 14 January 2009 (*issue date*), Alice (*author*)—i.e., the QSN member denoted by IRI http://www.qplus.net/alice—stated that all the resources hosted by example.org (*IRI pattern*) are safe for children (*description*). The message to be returned to the end user when accessing such resources is: "Everything on example.org is safe for children" (*summary*). This claim is valid from 1 January 2009 until 13 January 2010 (*validity period*).

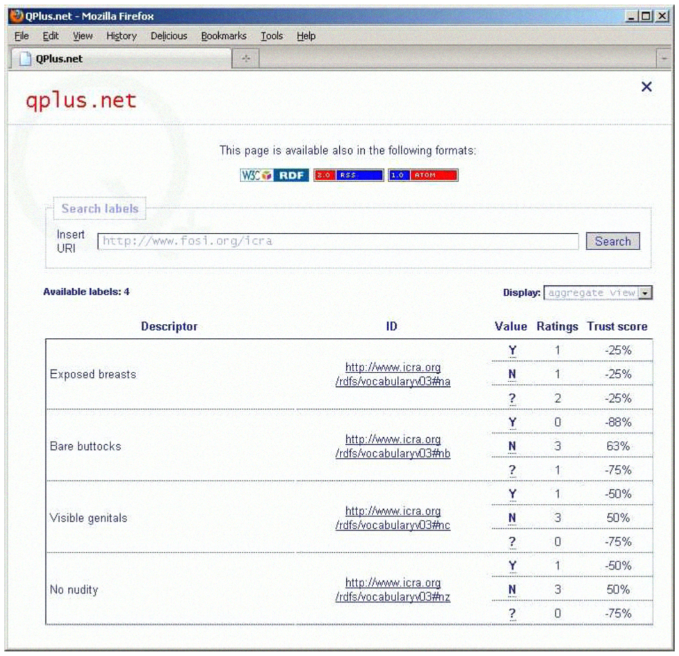
Labels are a feature which is quite similar to the one provided by social tagging services. However, differently from existing social tagging services (with the notable exception of MovieLens [17]), the QSN provides its members also the ability of expressing their dis/agreement about the existing labels, through *ratings*. A QSN rating is expressed by a rational number in the range $[-1..+1]$, and it includes also the ID of its author and its creation date.

One of the main purposes of the QSN is to use collaborative labeling and rating in order to assess labels' trustworthiness. As already mentioned in Section III, to achieve this, the descriptors contained in labels applying to a given resource, and the corresponding ratings, are aggregated and returned to the end user, along with a score denoting how much each descriptor can be trusted. As it will explained in Section V, label aggregation is exploited to evaluate the quality of a resource against the user preferences specified by a QSN member. However, it is not used only for this purpose. Actually, label aggregation is one of the most valuable services provided by the QSN, since it corresponds to the collective, weighted opinion of the whole QSN community about a given resource. For this reason, in the QSN, label aggregation is a public service which can be accessed by any Web user and software agent. Note that this do not raise privacy issues, since aggregated labels are anonymized—i.e., they do not carry information which can be used to infer the identities of labels' authors. Moreover, in order to increase as much as possible the effective exploitation of such service, the results of label aggregation are available in different formats. Currently, the supported formats are XHTML (for Web users), SOAP (for Web services), RDF (for Semantic Web agents), RSS and ATOM (for feed aggregators). An example of label aggregation represented in XHTML format is depicted in Figure 5.

As far as the computation of the aggregated labels' trustworthiness is concerned, we have decided to make the QSN as independent as possible from a specific algorithm. The reason is that, since the QSN can be run by different communities with different characteristics and purposes, administrators should be provided the ability of choosing the trust computation algorithm which is more suitable to their purposes. Currently, we support two different trust computation strategies. According to the former strategy, descriptors' trustworthiness is computed based only on their occurrences and on the associated ratings. By contrast, the latter takes into account also the *reputation* of labels and rating authors to weight the trust values obtained by the former trust computation strategy. Such reputation is determined based on the ratings specified in the QSN on the labels of a given QSN member. The support to more sophisticated trust computation algorithms is one of the issues we plan to address in future work (see Section VIII).

```
<powder xmlns="http://www.w3.org/2007/05/powder#"
     xmlns:qplus="http://www.qplus.net/voc#">
  <attribution>
    <issuedby src="http://www.qlus.net/alice"/>
    <issued>2009-01-14T00:00:00</issued>
    <validfrom>2009-01-14T00:00:00</validfrom>
    <validuntil>2010-01-13T00:00:00</validuntil>
  </attribution>
  <dr>
    <iriset>
      <includehosts>example.org</includehosts>
    </iriset>
    <descriptorset>
      <displaytext>Everything on example.org is safe for
           children</displaytext>
      <qplus:childSafe>true<qplus:childSafe>
    </descriptorset>
  </dr>
</powder>
```

Fig. 4. A QSN label represented as a POWDER document


QPlus.net - Mozilla Firefox
File Edit View History Delicious Bookmarks Tools Help
QPlus.net
qplus.net
This page is available also in the following formats:
RDF
RSS
ATOM
Search labels
Search

Insert URI: http://www.fosi.org/icra

Available labels: 4

Display: aggregate view

Descriptor	ID	Value	Ratings	Trust score
Exposed breasts	http://www.icra.org/rdfs/vocabularyv03#na	Y	1	-25%
		N	1	-25%
		?	2	-25%
Bare buttocks	http://www.icra.org/rdfs/vocabularyv03#nb	Y	0	-88%
		N	3	63%
		?	1	-75%
Visible genitals	http://www.icra.org/rdfs/vocabularyv03#nc	Y	1	-50%
		N	3	50%
		?	0	-75%
No nudity	http://www.icra.org/rdfs/vocabularyv03#nz	Y	1	-50%
		N	3	50%
		?	0	-75%

Fig. 5. A screenshot of the Web interface generated by the QUI, displaying the aggregate view of the labels concerning Web site http://www.fosi.org/icra

V. USER PREFERENCES

The purpose of user preferences (UPs) is to automatically perform an evaluation on the quality of a given resource, based on the requirements specified by a given QSN member. Such requirements are expressed by putting constraints on the resources' description stored by the QSN. The QSN members a UP applies to are denoted either by their identifiers or their attributes. Based on them, the UP engine (UPE) will return the user agent (UA) the *action* to be performed. Such action includes: (a) a notification about the quality of the resource, denoting whether it can be safely used, and, additionally, (b) a directive stating whether access to the resource must be blocked or granted.

A UP consists of the following main components:

- the *UP author* upa;
- the set M of QSN members the UP applies to, denoted by constraints on either the IRIs of their FOAF profiles, stored by the QSN (see Section III), or the contained descriptors;
- the set R of resources the UP applies to, denoted by constraints on either their IRIs or the descriptors derived from the associated POWDER documents;
- the *action* to be performed by the UA whenever a QSN member $m \in M$ requests access to a resource $r \in R$; the *UP action* consists in turn of the following main components:
  - a *quality score* $q \in [-1.. + 1]$, denoting whether r can be safely used by m;
  - an *access directive* d, denoting whether access to r must be granted or denied;
  - a message msg providing a human-readable explanation of the decision taken.

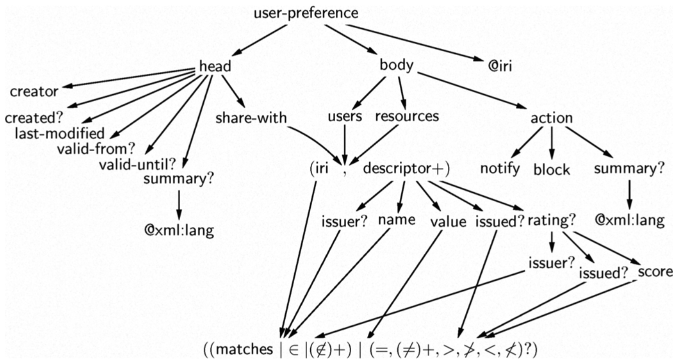


Fig. 6. Graphical representation of the UP syntax. In the figure, symbol '?' denotes that a given component is optional, whereas symbol '+' that the corresponding component must occur at least one time. Components associated with neither '?' nor '+' must occur exactly one time. Finally, symbol '|' is used to denote alternative components, whereas we use "(" and ")" to group elements.

Based on such syntax, the informal semantics of a UP can be described by the following statements:

> QSN member upa states that whenever a QSN member in M requests access to one of the resources in R, a quality score q must be returned, and access to r must be granted/denied.

In order to enforce UPs, it is now necessary to translate such informal definition of the UP syntax and semantics into a formal one. For this purpose, we can use *rule languages* and *systems* [18] both for UPs' representation and enforcement. An example of possible candidates are the Protune [19] and WIQA [20] frameworks, or N3Logic [21]. However, it is important to note that rule languages have an expressiveness which is far higher than the one required by our UPs, and they are not necessarily the best solution, in terms of efficiency. For this reason, we think it is preferable to design a UP language which could be efficiently processed. Based on such considerations, we have taken the decision of separating the syntactical from the semantic level of UPs.

As far as UP syntax is concerned, we have decided to use an XML-based format, since such solution grants two main advantages. First, XML is a widely used data interchange format: representing UPs in XML will then grant UPs' interoperability across different systems. Second, XML Schema [22], [23] allows a precise definition of the characteristics and structure of the components of an XML document—i.e., it can be used to define a formal syntax.

Figure 6 provides a graphical representation of UP syntax. As shown in the figure, the root element of a UP is user-preference, which takes two mandatory child elements, namely head and body, plus a required attribute, iri, which associates the UP with an IRI.

Element head is used to specify metadata concerning the UP itself, i.e., the UP author (element creator), denoted through the IRI of his/her FOAF profile, when the UP has been created (element created), the last modified date (element last-modified), the validity period (elements valid-from and valid-until), and a human-readable summary of the UP (element summary). Elements creator and last-modified are mandatory. Another mandatory child is element share-with, which specifies the QSN members who are authorized to reuse the UP. Depending on whether such members are denoted by the IRIs of their FOAF profiles or their properties, element share-with takes as child either the iri or descriptor elements. The characteristics of such elements are illustrated later in this section.

By contrast, element body specifies through its mandatory children users, resources, and action, the basic components of a UP. Element action has two mandatory children, namely, elements notify and block, plus an optional one, namely, element summary. Element notify takes as value a number in the range $[-1..+1]$, denoting the quality score to be returned, whereas element block denotes through a Boolean value whether access must be granted (FALSE) or denied (TRUE). Finally, element summary provides the human-readable description of the decision taken by the system based on the UP. Elements users and resources denote, respectively, the set M of QSN members and the set R of resources the UP applies to. Both elements share the same children. More precisely, the iri element is used to denote QSN members / resources based on their IRIs, whereas descriptor through their properties. The children of element iri correspond to a set of comparison operators. More precisely, element matches takes as value a regular expression, whereas elements one-of and not-one-of, which take as value a space-separated list of IRIs, correspond to set operators $\in$ and $\notin$, respectively.

The children of element descriptor are used to put constraints on the different components of a descriptor. Elements name, value, issuer, and issued take as children a set of elements corresponding to comparison operators. The last child of descriptor, i.e., element rating, denotes the trust score of the descriptor(s) denoted by the pair name-value, and, possibly, the ratings' authors and issue date. Element rating takes a mandatory child element, score, plus two optional ones, i.e., issuer and issued. Element score takes as children the same elements of issued.

Based on the formal UP syntax described so far, the semantics of a UP can be expressed through the following rule:

$$\forall xy, x \in M \wedge y \in R \rightarrow return(a)$$

where M and R correspond to the sets of QSN members and resources, respectively, denoted by the XML elements users and resources in a UP, whereas a denotes the action to be performed as specified in the XML element action of the UP. As far as element head is concerned, each of its children will correspond to a predicate having the rule above as subject.

Figure 7 reports an example of UP specification. The meaning of such UP can be summarized as follows:

```
<user-preference xmlns="..."                                                    1
    iri="http://www.qplus.net/alice/ups.xml#childSafe">
  <head>                                                                         2
    <creator>http://www.qplus.net/alice</creator>                                3
    <last-modified>2009-04-01T00:00:00</last-modified>                           4
    <summary xml:lang="en">This user preference specifies                         5
        the requirements to be satisfied by a resource to
        be considered as safe for children</summary>
    <share-with>                                                                 6
      <iri>                                                                      7
        <one-of>http://www.qplus.net/alice</includehosts>                        8
      </iri>                                                                     9
    </share-with>                                                               10
  </head>                                                                       11
  <body>                                                                        12
    <users>                                                                     13
      <iri>                                                                     14
        <one-of>http://www.qplus.net/alice</includehosts>                       15
      </iri>                                                                    16
    </users>                                                                    17
    <resources>                                                                 18
      <descriptor>                                                              19
        <name>                                                                  20
          <one-of>http://www.qplus.net/voc#childSafe</one-of>                   21
        </name>                                                                 22
        <value>                                                                 23
          <equal-to>true</equal-to>                                             24
        </value>                                                                25
        <rating>                                                                26
          <score>                                                               27
            <not-less-than>0.8</not-less-than>                                  28
          </score>                                                              29
        </rating>                                                               30
      </descriptor>                                                             31
    </resources>                                                                32
    <action>                                                                    33
      <notify>1</notify>                                                        34
      <block>false</block>                                                      35
      <summary xml:lang="en">The resource is safe for                           36
          children.</summary>
    </action>                                                                   37
  </body>                                                                       38
</user-preference>                                                              39
```

Fig. 7. An example of UP

- The UP has been specified by Alice (i.e., the QSN member denoted by IRI http://www.qplus.net/alice), and last modified on 1 April 2009 (lines 2-6).
- Alice is the only QSN member the UP applies to (lines 8-11).
- The UP applies to all the resources labeled with the descriptor childSafe, defined in the vocabulary identified by IRI http://www.qlus.net/voc#, having value TRUE, and a trust score of at least 80% (lines 13-27).
- The action to be performed requires (a) to return a 100% quality score, (b) to grant access to the resource, and (c) to display message "The resource is safe for children" (lines 28-32).

VI. User Preference Enforcement

The UP enforcement procedure consists of the following steps:

1) A given QSN member m submits an access request to resource r through a UA.
2) The UA sends a message (m, r) to the UPE.
3) The UPE retrieves from the QSN all the information available about m, which is then used to query the UP Base (UPB) in order to obtain the set UP of UPs applying to m according to the IRI and descriptors constraints specified in such UPs.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8368
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8368

4) The UPE then considers each $up \in UP$ in order to verify whether it applies or not to r, based on the IRI and descriptor constraints in up. The UPs not applying to r are removed from UP.
5) The UPE returns the set of UPs in UP, applying to both m and r.

The last step of UP enforcement is performed client-side by the UA, which is in charge of returning the end user the action specified by the UPs in UP. In case $|UP| = 1$ the UA performs the action specified by the returned UP. However, it may be often the case that no UP applies. In order to deal with such a situation, we have adopted the following approach. By default, the UA notifies the end user that no UP applies, and then the quality of the resource cannot be evaluated. In addition, the end user is given the ability to configure the UA to perform a *default action*, which can be customized by the end user him/herself.

Moreover, we have to deal also with situations where the UPE returns multiple, *conflicting* UPs. As an example, suppose that there exist two UPs up, up', applying to the same QSN member m and resource r. Suppose now that up gives a $+1$ quality score to r, and states that access to r must be granted. By contrast, up' gives a -1 quality score, and states that access must be denied. In such a case, we say that up, up' are in conflict. Such issue is addressed as follows. By default, the UA simply notifies the end user that there exist conflicting UPs. In addition, the end user can configure the UA to determine the prevailing UP based on a conflict resolution mechanism, based on parameters specified by the end user him/herself. More precisely, the end user that the prevailing UP is the one with the lowest or highest quality score and/or with an access directive requiring to deny or grant access to the requested resource.

The diagram in Figure 8 summarizes the procedure carried out by the UA in order to enforce the default action and the conflict resolution mechanism.

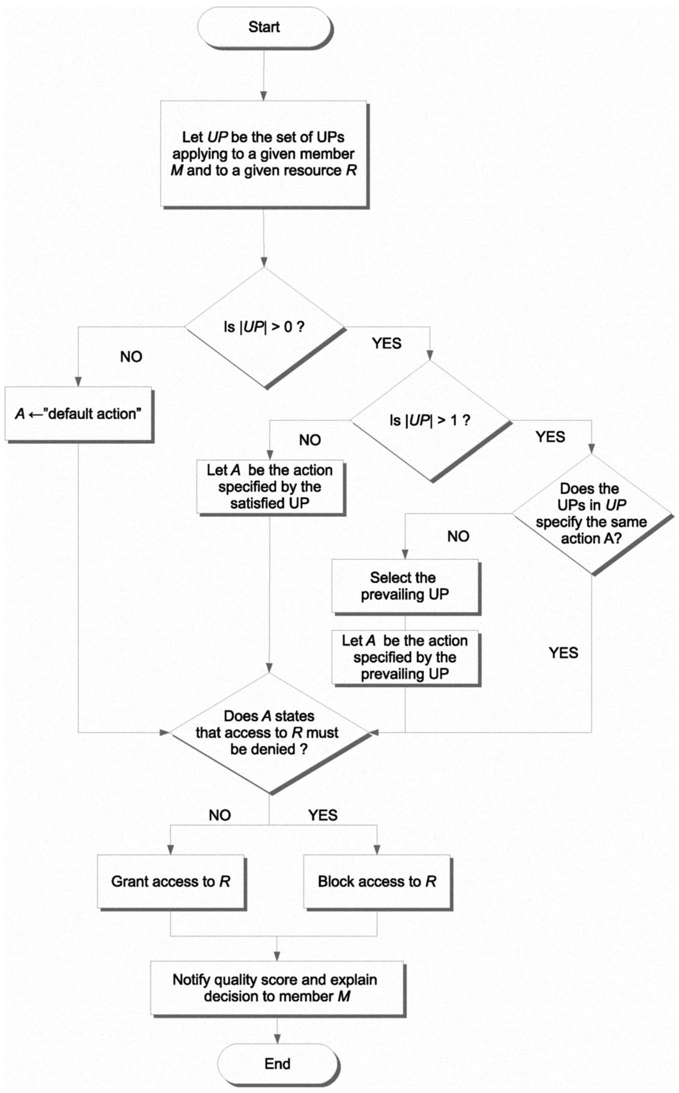

Start
Let UP be the set of UPs applying to a given member M and to a given resource R
Is $|UP| > 0$?
NO
YES
$A \leftarrow$"default action"
Is $|UP| > 1$?
NO
YES
Let A be the action specified by the satisfied UP
Does the UPs in UP specify the same action A?
NO
Select the prevailing UP
Let A be the action specified by the prevailing UP
YES
Does A states that access to R must be denied ?
NO
YES
Grant access to R
Block access to R
Notify quality score and explain decision to member M
End

Fig. 8. UP evaluation procedure

VII. Prototype Implementation and Use Case

A first implementation of the QSN has been carried out in the framework of the QUATRO Plus project, and then applied to a specific use case. More precisely, the QSN has been built on top of the PostgreSQL DBMS, which is in charge not only of storing QSN data, but also to enforce the basic operations concerning trust computation, labels' aggregation, and UP enforcement. We have adopted such an approach since a DBMS grants more efficiency, at least in terms of memory usage, when processing a huge amount of data, and because it offers built-in functionalities (such as materialized views), to efficiently implement operations which are computationally costly. By contrast, the QUI and the SOAP interfaces to QSN services have been built in PHP, which has the advantage of providing native support to the most diffused DBMSs, and to XML, XSLT, and SOAP. In addition, it is supported by most of the existing operating systems and Web servers. Thanks to these features, it is possible to build a QSN installation package which can be easily installed and configured on most of the existing platforms. Finally, the QSN prototype includes a Web interface, generated by the QUI according to the strategy described in Section III.

The QSN prototype has then be revised and extended in order to address a specific use case. More precisely, three QSN instances have been installed for the Greek, Dutch, and Czech Youth Panels of the Safer Internet European Programme,[5] with the purpose of testing its efficiency, effectiveness, and usability. As far as label specification is concerned, each QSN instance makes use of a set of descriptors defined by the corresponding Youth Panel.

The use case is still in its testing phase, so we cannot yet provide experimental results. However, based on the requirements of the Youth Panels and the preliminary feedback we

[5]Safer Internet Youth Panels are typically groups of 5-30 teenagers, established by national Safer Internet Centres, which provide feedback about Safer Internet initiatives. For more details, we refer the reader to: http://ec.europa.eu/information_society/activities/sip/projects/centres/panels.

have received so far, the QSN prototype has been refined as follows:

- QSN services and the QSN database schema have been revised in order to improve efficiency.
- Localized versions of the QUI have been developed, which is now available in Czech, Dutch, English, Greek, and Italian.
- Configuration options have been provided for the QUI, thanks to which it is possible to choose the default language and the theme to be used by the generated Web interface.

VIII. Conclusions and Future Work

In this paper, we have presented a collaborative social networking environment, the QSN, developed in the framework of the QUATRO Plus EU project, which supplies Web access personalization by giving its members the ability to specify user preferences. User preferences are a means to assess the quality of Web resources based on the labels and ratings collected by the network. Besides its main features, we have described its prototype implementation and its application to a use case scenario involving three Safer Internet Youth Panels.

Future work will span over different directions. Besides revising the prototype based on the test results obtained from the current use case scenario, we plan to extend it in order to support further and more sophisticated trust computation algorithms. For this purpose, we will investigate existing trust algorithms proposed for reputation systems [24], recommender systems [25], and social networks (e.g., [26], [27]), in order to verify whether and how they can be revised in order to meet the requirements of the QSN. In addition to this, we plan to develop a new version of the QSN, which can be plugged in into existing online communities through standard APIs, in order to extend them with the support to Web access personalization. Thanks to this, the QSN would become a social application, able to reuse existing social data, may them be personal relationships or social tags.

Acknowledgment

The work reported in this paper is partially funded by the European Community under the QUATRO Plus project (SIP-2006-211001).

References

[1] S. A. Golder and B. A. Huberman, “The structure of collaborative tagging systems,” *Comput. Res. Repository*, vol. abs/cs/0508082, 2005. [Online]. Available: http://arxiv.org/abs/cs/0508082

[2] J. Voß, “Tagging, folksonomy & Co – Renaissance of manual indexing?” *Comput. Res. Repository*, vol. abs/cs/0701072, 2007. [Online]. Available: http://arxiv.org/abs/cs/0701072

[3] B. Carminati, E. Ferrari, and A. Perego, “Combining social networks and Semantic Web technologies for personalizing Web access,” in *CollaborateCom 2008. Revised Selected Papers*, ser. LNICST, vol. 10. Springer, 2009, pp. 126–144.

[4] M. J. Dürst and M. Suignard, “Internationalized resource identifiers (IRIs),” Internet Engineering Task Force, IETF RFC 3987, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3987.txt

[5] T. Berners Lee, R. T. Fielding, and L. Masinter, “Uniform resource identifier (URI): Generic syntax,” Internet Engineering Task Force, IETF RFC 3986, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3986.txt

[6] T. Berners Lee, L. Masinter, and M. McCahill, “Uniform resource locators (URL),” Internet Engineering Task Force, IETF RFC 1738, Dec. 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1738.txt

[7] R. Moats, “URN syntax,” Internet Engineering Task Force, IETF RFC 2141, May 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2141.txt

[8] P. Archer, K. Smith, and A. Perego, “Protocol for Web description resources (POWDER): Description Resources,” World Wide Web Consortium, W3C Recommendation, Sep. 2009. [Online]. Available: http://www.w3.org/TR/powder-dr/

[9] P. Archer, A. Perego, and K. Smith, “Protocol for Web description resources (POWDER): Grouping of resources,” World Wide Web Consortium, W3C Recommendation, Sep. 2009. [Online]. Available: http://www.w3.org/TR/powder-grouping/

[10] S. Konstantopoulos and P. Archer, “Protocol for Web description resources (POWDER): Formal semantics,” World Wide Web Consortium, W3C Recommendation, Sep. 2009. [Online]. Available: http://www.w3.org/TR/powder-formal/

[11] P. Archer, E. Ferrari, V. Karkaletsis, S. Konstantopoulos, A. Koukourikos, and A. Perego, “QUATRO Plus: Quality you can trust?” in *ECWS 2009 SPOT Workshop*, ser. CEUR Workshop Proceedings, vol. 447. CEUR-WS.org, 2009. [Online]. Available: http://CEUR-WS.org/Vol-447/paper1.pdf

[12] D. Brickley and L. Miller, “FOAF Vocabulary Specification v0.91,” Namespace Document, Nov. 2007. [Online]. Available: http://xmlns.com/foaf/0.1/

[13] I. Davis and E. Vitiello, Jr, “RELATIONSHIP: A vocabulary for describing relationships between people,” Namespace Document, May 2009. [Online]. Available: http://purl.org/vocab/relationship

[14] J. Clark, “XSL transformations (XSLT) – version 1.0,” World Wide Web Consortium, W3C Recommendation, Nov. 1999. [Online]. Available: http://www.w3.org/TR/xslt

[15] Mozilla Developer Center, “XUL reference,” Technical Specification, Aug. 2009. [Online]. Available: https://developer.mozilla.org/en/XUL_Reference

[16] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible markup language (XML) 1.0 (fifth edition),” W3C Recommendation, Nov. 2008. [Online]. Available: http://www.w3.org/TR/xml/

[17] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl, “tagging, communities, vocabulary, evolution,” in *CSCW 2006*. ACM Press, 2006, pp. 181–190.

[18] G. Antoniou, M. Baldoni, P. A. Bonatti, W. Nejdl, and D. Olmedilla, “Rule-based policy specification,” in *Secure Data Management in Decentralized Systems*, ser. Advances in Information Security, T. Yu and S. Jajodia, Eds. Springer, 2007, vol. 33, pp. 169–216.

[19] P. A. Bonatti and D. Olmedilla, “Driving and monitoring provisional trust negotiation with metapolicies,” in *POLICY 2005*. IEEE CS Press, 2005, pp. 14–23.

[20] C. Bizer and R. Cyganiak, “Quality-driven information filtering using the WIQA policy framework,” *J. Web Semant.*, vol. 7, no. 1, pp. 1–10, Jan. 2009.

[21] T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler, “N3Logic: A logical framework for the World Wide Web,” *Theory Pract. Log. Program.*, vol. 8, no. 3, pp. 249–269, May 2008.

[22] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, “XML schema part 1: Structures – second edition,” World Wide Web Consortium, W3C Recommendation, Oct. 2008. [Online]. Available: http://www.w3.org/TR/xmlschema-1/

[23] P. V. Biron and A. Malhotra, “XML schema part 2: Datatypes – second edition,” World Wide Web Consortium, W3C Recommendation, Oct. 2008. [Online]. Available: http://www.w3.org/TR/xmlschema-2/

[24] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[25] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[26] P. Avesani, P. Massa, and R. Tiella, “A trust-enhanced recommender system application: Moleskiing,” in *ACM SAC 2005*. ACM Press, 2005, pp. 1589–1593.

[27] J. A. Golbeck, “Generating predictive movie recommendations from trust in social networks,” in *iTrust 2006*, ser. LNCS, vol. 3986. Springer, 2006, pp. 93–104.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8368
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8368