A Model to Support Multi-Social-Network Applications

Francesco Buccafurri, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera

DIIES, University of Reggio Calabria
Via Graziella, Località Feo di Vito
89122 Reggio Calabria, Italy
{bucca,lax,s.nicolazzo,a.nocera}@unirc.it

Abstract. It is not uncommon that people create multiple profiles in different social networks, spreading out over them personal information. This leads to a multi-social-network scenario where different social networks cannot be viewed as monads, but are strongly correlated to each other. Building a suitable middleware on top of social networks to support internetworking applications is an important challenge, as the global view of the social network world provides very powerful knowledge and opportunities. In this paper, we do a first important step towards this goal, by defining and implementing a model aimed at generalizing concepts, actions and relationships of existing social networks.

1 Introduction

Online social networks have quickly become one of the most important communication media able to attract people and companies and to support a large number of applications in various contexts. The power of online social networks mainly derives from the fact that people are directly connected to each other, thus enabling the word-of-mouth effect (as in real social networks) on which information diffusion and people influencing is based.

In the dynamics of this phenomenon, an important role is played by membership overlap among different online social networks, thanks to which social networks are interconnected to each other. Indeed, when a user joins different networks, he plays the role of *bridge* for the information coming from one network to another. This happens because the information received in a social network can be manually posted (and thus diffused) on the other social network. Sometimes, automatic functions are enabled, such as auto posting from Twitter to Facebook or from Twitter to Facebook, so that each received tweet is posted on the Facebook's wall and vice versa.

The recent literature has highlighted that the aforementioned multiple-social-network perspective opens a lot of new problems in terms of analysis [7,3,5] but also new opportunities from the application point of view [4]. For example, consider the possibility of building the complete profile of users by merging all the information they spread out over the distinct social networks they belong to. This gives a considerable added value to market analysis and job recruitment

R. Meersman et al. (Eds.): OTM 2014, LNCS 8841, pp. 639–656, 2014.

strategies, as membership overlap among social networks is often an expression of different traits of users personality (sometimes almost different identities).

Again, consider the field of identity management [10,14,32]: To trust user's identity or to identify fake profiles a cross check involving different social networks can be used.

From the above observations, it clearly follows that even though each single social network is an extraordinary source of knowledge, the information power of the social-network Web can be considerable increased if we see it as a huge global social network, composed of autonomous components with strong correlation and interaction.

However, despite the conceptual uniformity of the social-network universe, in terms of structure, basic mechanisms, main features, etc., each social network has in practice its own terms, resources, actions. This is a strong handicap for the design and implementation of applications enabling internetworking functions among different social networks, and, then, for the achievement of the above goal. As a consequence, building a suitable middleware on top of social networks to support internetworking applications is an important challenge, because the global view of the social network world can provide very powerful knowledge and opportunities.

In this paper, we do a first important step towards this goal, by defining and implementing a model aimed at generalizing concepts, actions and relationships of existing social networks.

This paper is organized as follows. In the next section, we describe the reference scenario. We give a formal definition of our graph-based conceptual model in Section 3. We implement the model by defining suitable mappings among concepts and social networks functionalities in Section 4. In Section 5, we show how our model can be profitably applied to some very relevant applications in the context of social network analysis. In Section 6, we survey the related work. Finally, we draw our conclusions and make some reasonings on possible future work in Section 7.

2 Reference Scenario

This section provides a description of the reference scenario. Moreover, it helps the reader to recall the main features of social networks. Such features are modeled in this paper.

An Online Social Network (OSN) provides powerful technical features to make communication among users easy. Its backbone consists of public profiles, which collect personal information and interests, and an articulated list of friends who are other users of the system. When a user joins a social network, she has to fill her own profile with descriptors, such as age, location, interests, photos and multimedia contents. Moreover, a OSN describes entities and connections among them. Entities are often individuals who are connected to each other by personal relationships, interactions, or information flows. The collection of friend nodes is not simply a list of close ties. It represents a microcosm inside the social network, where each node can interact with the other ones. Because a friend list is visible to everyone, users can exploit it to trace friend links. A new participant can find and register a new friend using the friend lists of the other users.

Profiles and friend lists are only two key features of social networks. The third feature is a functionality allowing users to write comments about other users. These comments concerning a user profile are displayed prominently and are visible to anyone who can access that profile (i.e, to anyone who is in the friend list of the profile owner).

The three features (profiles, friends lists and comments) represent the basic structure of a social network site. Moreover, all social networks have a set of basic functionalities which are considered essential to qualify them as a social networking service. These functionalities are:

- the ability to set up and customize a personal profile by simple forms;
- an utility that allows members to comment others activities;
- a system that allows users to make a granular control of shared information (privacy settings);
- the ability to block an unwanted member in order to exclude him from the friend list;
- a personal home page which can contain personal information, notes and individual picture albums;
- the ability to own, form or be member of a group or a community within the network;
- the ability to include new "social applications", or gadgets, which emphasize the information spreading;
- instantaneous messages;
- photo tagging tools;
- notification via SMS;
- photo and video sharing.

A *multi-social network* extends the concept of single social network by taking into account the multiplicity of social networks and their interconnections through *me* edges, by means of which a user may link other (social network) accounts she has.

3 The Conceptual Model

To model at an abstract level our multi-social-network scenario we use a graph. The set of nodes is partitioned into three disjoint sets P, R, and B, which correspond to the set of social profiles, the set of resources, and the set of bundles (which are resource containers), respectively.

An element of P models the profile of a user on a social network. It consists in a tuple ⟨url, socialNetwork, screen-name, [personalInformation], [picture]⟩, where url is the Web address that identifies and localizes the profile, socialNetwork is the commercial name of the social network which the profile belongs to, screen-name is the name chosen by the user who registered the profile to appear in the home-page of the profile or when posting a resource, and, finally, personalInformation and picture are the information and the image which the user inserted as related to the profile. The two last elements of the tuple are optional (i.e., they can be null).

The set R models resources of the Web or created by users. A resource is represented by a tuple ⟨url, type, [description], [date]⟩, where url is the Web address to access the resource, type indicates the type of the resource content, and finally, description and date, which are optional, represent the string, inserted by the user who published the resource, describing the resource and the publishing date, respectively. For example, the most viewed video on YouTube [1] is a resource represented as ⟨'https://www.youtube.com/watch?v=9bZkp7q19f0', 'video/mp4', 'PSY - GANGNAM STYLE', '07/15/2012'⟩.

Our model includes the *bundle* set B. Indeed, commonly users do not handle a single resource, but most of the actions they do (e.g., publishing or sharing) involve more resources simultaneously. For example, a user can publish more photos or videos, can include a comment, and so on. In our model, we include all resources handled simultaneously by a user in a *bundle*. A bundle is represented by a tuple ⟨uri, [description], [date]⟩, where uri is the identifier of the bundle, description, which is optional, is the string chosen by the user to be shown with those resources and, finally, date represents the publishing date. As we will see next, we represent the inclusion of a resource into a bundle by means of *containing* edges.

In our model, relationships among profiles, resources and bundles are represented by direct edges of a graph. According to the specifics reported in Section 2, the set E of edges is partitioned into 8 disjoint sets, named F, M, Pu, S, T, Re, L, and Co.

The *follow* edge set $F \subseteq E = \{p_s, p_t \mid p_s, p_t \in P\}$ denotes that in the (source) profile p_s, it has been declared a certain type of relationship towards the (target) profile p_t. This kind of edge models different relationships. For example, on Facebook or Flickr, it models friendship, on LinkedIn, job contacts, and, on Twitter, *followers*. Observe that, typically, this kind of relationship occurs between users of the same social network, because it is presumable that a social network does not have interest in promoting links to profiles of another (competitor) social network.

The *me* edge set $M \subseteq E = \{p_s, p_t \mid p_s, p_t \in P\}$ denotes that the user with profile p_s has declared in this profile to have a second profile p_t. This edge allows a user to provide a link to its profile (typically) on a different social network or (sometimes) on the same social network (as a sort of alias).

The *publishing* edge set $Pu \subseteq E = \{p_s, b_t \mid p_s \in P, b_t \in B\}$ denotes that the user with profile p_s has published in this profile a bundle b_t. This edge models one of the typical actions a user does when enriches his/her profile by publishing resources.

The *shared* edge set $S \subseteq E = \{b_s, b_t \mid b_s, b_t \in B\}$ denotes that the bundle b_s (published by a user) is derived from an already published bundle b_t. This type of edge is used when a user shares an existing bundle. Indeed, this action is represented by two edges: a publishing edge (as described before) and a shared edge from the new bundle to the existing one.

The *tagging* edge set $T \subseteq E = \{p_s, br_t, w \mid p_s \in P, br_t \in B \cup R$ and w is a word$\}$, denotes that the user with profile p_s assigned the word w to describe a bundle or a resource br. By means of the tag mechanism, users contribute to resource labelling, which is necessary to carry out several actions on resources, such as searching or classification.

The *referencing* edge set $Re \subseteq E = \{b_s, p_t \mid b_s \in B, p_t \in P\}$ models the fact that a bundle b_s includes a reference to the profile p_t. For example, this occurs when a *tweet* includes a user account name.

The *like* edge set $L \subseteq E = \{p_s, pbr_t \mid p_s \in P, pbr_t \in B \cup R \cup P\}$ models the information that a user with the profile p_s expressed a preference/appreciation for a bundle, a resource or another user profile pbr_t.

The *containing* edge set $Co \subseteq E = \{b_s, r_t \mid b_s \in B, r_t \in R\}$ models the fact that a bundle b_s contains the resource r_t. For example, when a user publishes a photo p and includes a comment c, this action is modeled by creating a bundle b with a description c, a resource p, and finally, a *containing* edge from b to p.

After defining the conceptual model, we will show how to practically map real-life data from social networks to each component of the model, in such a way to build a data structure that can be used at application level (as we will show in Section 5).

4 Building the Model

Information necessary to build the model can be extracted from social networks via four technologies: (i) APIs provided by the social network; (ii) FOAF datasets; (iii) XFN microformat; and (iv) HTML parsing.

As for the first technology, social network APIs are a platform available for developers which allow the access to social-networks data so as to create applications on top of them. Usually, there are different kinds of APIs each providing specific services. Among them, the most commons are the REST API, the Search API and the Streaming API. Specifically, the REST APIs allow operations such as insert, update or deletion to be performed. The Search APIs, instead, are useful to query the database and, finally, the Streaming APIs are conceived for applications that need to receive real-time updates (such as, new posts or feeds).

The second possible strategy to extract information from social network relies on FOAF datasets. The FOAF project [2] focuses on the creation of a machine-readable ontology describing friendship relationships among users. FOAF data sources allow the representation of a whole social network without the need of a centralized database. In fact, by relying on this technology, it is possible to represent the information concerning a user account, along with the corresponding contacts and activities, through an RDF graph serialized as an XML document, according to the W3C RDF/XML syntax.

The third option makes use of XFN microformat. It allows for the representation of the kind of relationship existing between two user accounts. This is

```
{
  "id": "1587099156",
  "first_name": "Serena",
  "gender": "female",
  "last_name": "Nicolazzo",
  "locale": "en_GB",
  "name": "Serena␣Nicolazzo",
  "username": "serena.nicolazzo"
}
```


Listing 1.1. An example of the output of the Facebook Graph API.

obtained by empowering the set of values that the `rel` attribute of the HTML tag `<a>` (which represents a link) can assume. In our case, we focus on the value "me" (`rel='me'`) which indicates that the corresponding link represents a `me` edge.

The last data extraction strategy leverages on HMTL parsing. Processing HTML to obtain social data is the most intricate procedure. Parsing requires much time because it needs to analyze all context information from the page source code. It is a low-level way of dealing with social data. Because the code written depends on the HTML page structure, it is not stable (due to the frequent graphical changes). For this reason, this strategy needs continue maintenances. However, it remains a valid alternative when other more practical solutions (like APIs, for instance) are not available.

Now we will show some significant examples on how the information represented by our model are extracted from social networks.

As for the user profile P described in Section 3, it consists in a tuple ⟨url, socialNetwork, screen-name, personalInformation, picture⟩. For example, to extract the information to build the profile of a Facebook user, we use the `Graph APIs`, accessible through the url `http://graph.facebook.com/{user-id}` or `http://graph.facebook.com/{screen-name}`. The output of this API is a JSON file (see, for instance, Listing 1.1).

We can extract from this JSON the user id, her username (which correspond to our notion of screen-name) and her personal information like: first name, last name, gender, locale (chosen language). The field *url* available in our social profile object can be obtained as `http://www.facebook.com/{screen-name}`, whereas the field *picture* can be obtained by another call to the `Graph APIs`, specifically by accessing the url `http://graph.facebook.com/{user-id}/picture`.

Many social networks are equipped also with FOAF datasets. As an example, we show how the edges belonging to the set *follow* can be obtained for the social networks LiveJournal and Advogato. The FOAF datasets for both social networks are reachable through the specific URLs `http://{screen-name}.livejournal.com/data/foaf` (for LiveJournal) and `http://www.advogato.org/person/{screen-name}/foaf.rdf` (for Advogato).

An example of an XML serialization of a FOAF document is shown in Listing 1.2. In this document, the information needed to build an edge of the set *follow*

```
1 <?xml version='1.0'?>
2 <rdf:RDF
3   xml:lang="en"
4   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
6   xmlns:foaf="http://xmlns.com/foaf/0.1/"
7   xmlns:ya="http://blogs.yandex.ru/schema/foaf/"
8   xmlns:lj="http://www.livejournal.org/rss/lj/1.0/"
9   xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
10  xmlns:dc="http://purl.org/dc/elements/1.1/">
11  <foaf:Person>
12      <foaf:nick>antoninonocera</foaf:nick>
13      <foaf:name>real_name</foaf:name>
14      <foaf:openid rdf:resource="http://antoninonocera.livejournal.com/"/>
15      <foaf:weblog rdf:resource="http://antoninonocera.livejournal.com/"/>
16      <foaf:homepage rdf:resource="*****" dc:title=""/>
17      ...
18      <foaf:knows>
19         <foaf:Person>
20            <foaf:nick>contact1</foaf:nick>
21            <foaf:member_name>contact_real_name</foaf:member_name>
22            <foaf:tagLine></foaf:tagLine>
23            <foaf:image>http://l-userpic.livejournal.com/****/****</foaf:image>
24            <rdfs:seeAlso rdf:resource="http://contact1.livejournal.com/data/foaf"/>
25            <foaf:weblog rdf:resource="http://contact1.livejournal.com/"/>
26         </foaf:Person>
27      </foaf:knows>
28      ...
29  </foaf:Person>
30 </rdf:RDF>
```

Listing 1.2. An XML-serialized FOAF document.

```
1     <a class="0La␣url␣Xvc" href="http://www.youtube.com/channel/<screen-name>"
2     rel="me" target="_blank" title="<screen-name>"><Link_name></a>
```

Listing 1.3. An example of a *me* edge using XFN.

can be extracted from lines 11 to 29. Specifically, the element $\langle foaf : Person \rangle$ indicates the beginning of the portion of the document where information about a user, her contacts and, often, her activities are reported. The information about each contact is encoded as a $\langle foaf : Person \rangle$ nested inside a tag $\langle foaf : knows \rangle$.

Concerning the information about *me* edges, it can often be extracted through the XFN microformat. Some examples of social networks adopting this standard to represent *me* edges are About.me, Advogato, Boards, Facebook, Flickr, Google+, and Twitter. Listing 1.3 shows the code representing a *me* edge in the social network Google+. The code at line 2 represents the explicit declaration that the corresponding link encodes a relation of type *me*.

Another interesting example regards the extraction of the information needed to build a *publishing* edge. Consider the social network LinkedIn. It provides a search API, called Job Lookup API, to obtain information about jobs that can be accessed at the address `http://api.linkedin.com/v1/jobs/job_id:(id,company,posting-date)`. The XML output produced by the call to this API is reported in Listing 1.4. In this case, when a company proposes a new job position (publishing), we model this event by adding two objects: (i) a bundle and (ii) a *publishing* edge between the profile of the company and the bundle just created (see Section 3). As for the bundle, the field `uri` is mapped to the element

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <job>
3        <id>1511685</id>
4        <expiration-timestamp>1304030488000</expiration-timestamp>
5        <company>
6              <id>229433</id>
7              <name>Cloudera</name>
8        </company>
9        <position>
10             <title>Technical Writer</title>
11             <location>
12             <name>San Francisco Bay Area</name>
13             <country>
14                   <code>us</code>
15             </country>
16             </location>
17       </position>
18       <location-description>San Francisco or Palo Alto,CA</location-description>
19       <job-poster>
20             <id>hQ4ruu3J2q</id>
21             <first-name>Paul</first-name>
22             <last-name>Battaglia</last-name>
23             <headline>Technical Writer at Cloudera</headline>
24       </job-poster>
25  </job>
```


Listing 1.4. An example of the output of the LinkedIn Job Lookup API.

$< id >$ (line 3), whereas the field `description` is obtained from the elements `<company>` (lines 5-6), `<position>` (lines 9-17), and `<localition-description>` (line 18). The *publishing* edge is associated with the user profile whose identifier is specified by the element `<job-poster>` (lines 19-24) and has the new created bundle as target.

Now, consider the case of the publishing of a new tweet containing a resource and referencing another user in the Twitter social network. Our model represents this action by adding the following objects: (i) a bundle, (ii) a resource, (iii) a *publishing* edge, (iv) a *containing* edge, and (v) a *referencing* edge. In this case, all information required by our model is extracted from Twitter by means of the method *GET statuses/user_timeline* of the Twitter APIs. Listing 1.5 shows an example of the output of this API. Specifically, line 6 is the tweet identifier and is mapped to the field `uri` of the bundle. The bundle field `description` is obtained by suitably parsing lines 24-30. The bundle is linked to the publisher user by means of a *publishing* edge. As mentioned above, a new resource is added and associated with the bundle by means of a *containing* edge. Information needed to create the resource is extracted from lines 11-23. In particular, the field `url` is obtained from line 15, the field `type` from line 20, and, finally, the field `description` is extracted from lines 19, 20 and 21. The tweet in the example references another user and this action is modeled by adding a *referencing* from the bundle to the user specified on lines 9 and 10.

An important feature, common to almost all social networks, is the possibility to appreciate a resource or another user profile. In our model, this concept is represented by means of the *like* edge. Consider the social network About.me, in which a user is allowed to favor another user profile, thus making an "endorsement". Information about this action is obtained by calling the method `http://api.about.me/api/v2/json/user/view/<screen_name>`

```
{
  "coordinates": null,
  "favorited": false,
  "truncated": false,
  "created_at": "Wed␣Aug␣29␣17:12:58␣+0000␣2012",
  "id_str": "240859602684612608",
  "entities": {
        "hashtags": [ ],
        "user_mentions":[{"indices": [3, 10],"id_str": "<user_id>", "screen_name": "<user_screen_name>",
        "name": "<user_real_name>", "id": <user_id>}]
        "media": [{
                "id": 266031293949698048,
                "id_str": "266031293949698048",
                "indices": [17, 37],
                "media_url": "http://pbs.twimg.com/media/A7EiDWcCYAAZT1D.jpg",
                "media_url_https": "https://pbs.twimg.com/media/A7EiDWcCYAAZT1D.jpg",
                "url": "http://t.co/bAJE6Vom",
                "display_url": "pic.twitter.com/bAJE6Vom",
                "expanded_url":"http://twitter.com/BarackObama/status/266031293945503744/photo/1",
                "type": "photo",
                "sizes": {...}
        }]
   },
   "in_reply_to_user_id_str": null,
   "contributors": null,
   "text": "Introducing␣the␣Twitter␣Certified␣Products␣Program:␣https://t.co/MjJ8xAnT",
   "retweet_count": 121,
   "in_reply_to_status_id_str": null,
   "id": 240859602684612608,
   "geo": null,
   "retweeted": false,
   "possibly_sensitive": false,
   "in_reply_to_user_id": null,
   "place": null,
   "user": {...},
   "in_reply_to_screen_name": null,
   "source": "<a␣href=\"http://sites.google.com/site/yorufukurou/\"␣rel=\"nofollow\">YoruFukurou</a>",
   "in_reply_to_status_id": null
 }
```


Listing 1.5. An example of the output of the Twitter API method *user_timeline*.

of the About.me API. Listing 1.6 reports an example of the output of this method. The returned information has to be seen from the "caller" point of view (i.e., the authenticated user), therefore the line 21 indicates that the user *screen_name* is in the favourite list of the authenticated user. According to our model, a *like* edge from the authenticated user to the user with the given *screen_name* is created. A similar reasoning can be applied also for "g+1" of Google+ and "Like" of Facebook.

So far, we have seen how to extract information from different social networks and how to map them to the concepts defined in our model. Once this mapping has been done, a data-structure is obtained. It can be serialized using the XML language. In the following, we will show some details about the XML schema underlying our model.

Figures 1, 2 and 3 show the tree-based representation of our XML schema. The root element is *SocialGraph* and contains two unbounded sets of elements, namely *SocialNode* and *SocialEdge*. An element *SocialNode* is specialized in one of the following complex types: *SocialProfile*, *Resource*, or *Bundle*. The element *SocialEdge* is specialized in one of the following complex types: *Follow*, *Me*, *Publishing*, *Tagging*, *Shared*, or *Referencing*. Each complex-type in this XML is defined according to the corresponding objects defined in Section 3. Listing

```
{
    "status": 200,
    "profile": "http://about.me/<screen_name>",
    "user_name": "test␣account",
    "first_name": "test22",
    "last_name": "tester",
    "display_name": "test22␣tester",
    "header": "my␣headline",
    "bio": "test␣this␣is␣one!!!!",
    "background": "http://about.me/.../<screen_name>_1326415784_79.jpg",
    "mobile_background": "",
    "email_searchable": true,
    "email_public": false,
    "avatar": "http://about.me/.../<screen_name>_1325746595_83.jpg",
    "img_base_url": "http://about.me/.../thumbnail",
    "thumbnail_291x187": "http://about.me/.../291x187/<screen_name>.jpg",
    "thumbnail1": "http://about.me/.../803x408/<screen_name>.jpg",
    "thumbnail2": "http://about.me/.../260x176/<screen_name>.jpg",
    "thumbnail3": "http://about.me/.../198x134/<screen_name>.jpg",
    "thumbnail4": "http://about.me/.../161x109/<screen_name>.jpg",
    "is_fav": true
}
```


Listing 1.6. An example of the output of the About.me API method *view*.

1.7 reports a fragment of the XML schema whose tree-representation has been described above.

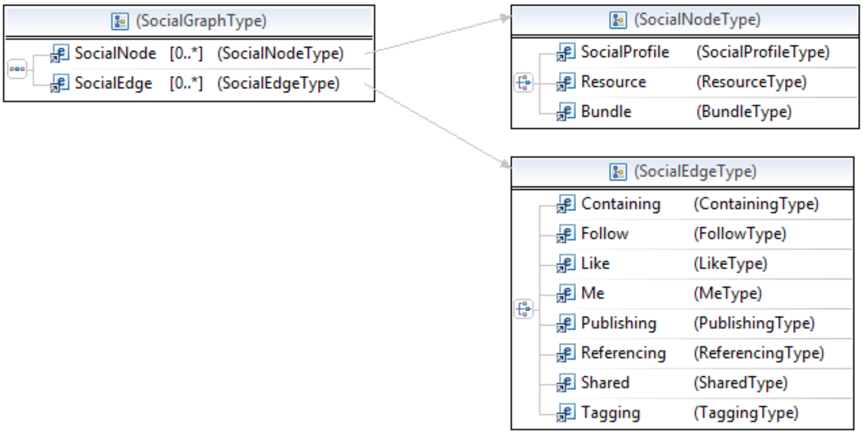

Fig. 1. A portion of the tree representation of our XML Schema (1/3)

We conclude this section by showing in Listing 1.8 an example of a fragment of an XML document derived from the XML Schema described above. Lines 6-20 show the definition of the Twitter profiles of two of the authors of this paper. In lines 39-44, a *follow* edge among them is defined. Lines 22-37 represent the definition of a new bundle and a new resource of type "photo" (line 33). The bundle is published by one of the authors of this paper and the *publishing* action

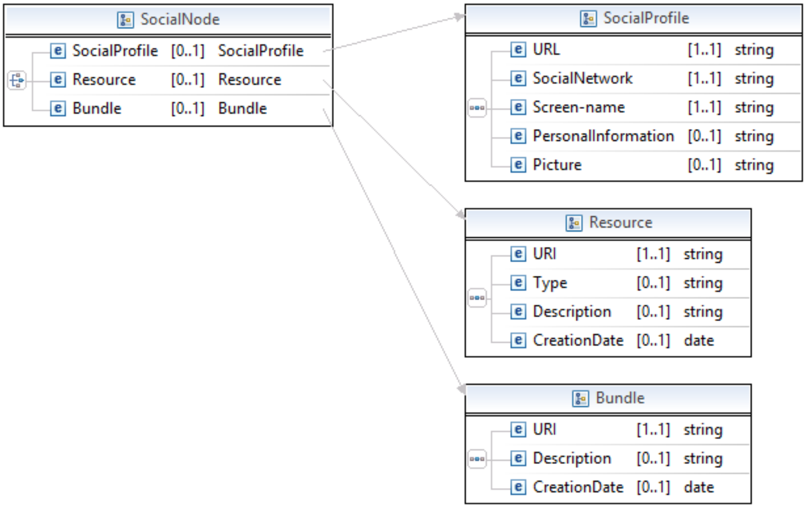

Fig. 2. A portion of the tree representation of our XML Schema (2/3)

is encoded in lines 46-51. The resource is contained in the bundle as shown in lines 53-58 with the definition of a *containing* edge. Finally, the information that one of the authors has two accounts in Twitter and Facebook is modeled in lines 60-74.

5 Case Study

In this section, we briefly describe how our model has been profitably applied to two applications very relevant in the context of social network analysis. The first application regards the extraction of information from a multi-social-network scenario, the second one concerns a particular analysis done on social network data.

It is well known that any analysis activity on social network users needs a preliminary task implementing the extraction of data from social networks. In the past, several visit strategies have been adopted, such as Breadth First Search [31], Random Walk [20] or Metropolis- Hastings Random Walk [27]. In all these cases, data analysis focused on a single social network and data extraction was a quite simple task because there was not the problem of receiving data from different sources.

When data extraction involves different social networks, having a model that is able to handle indifferently data from different social networks is a very useful tool.

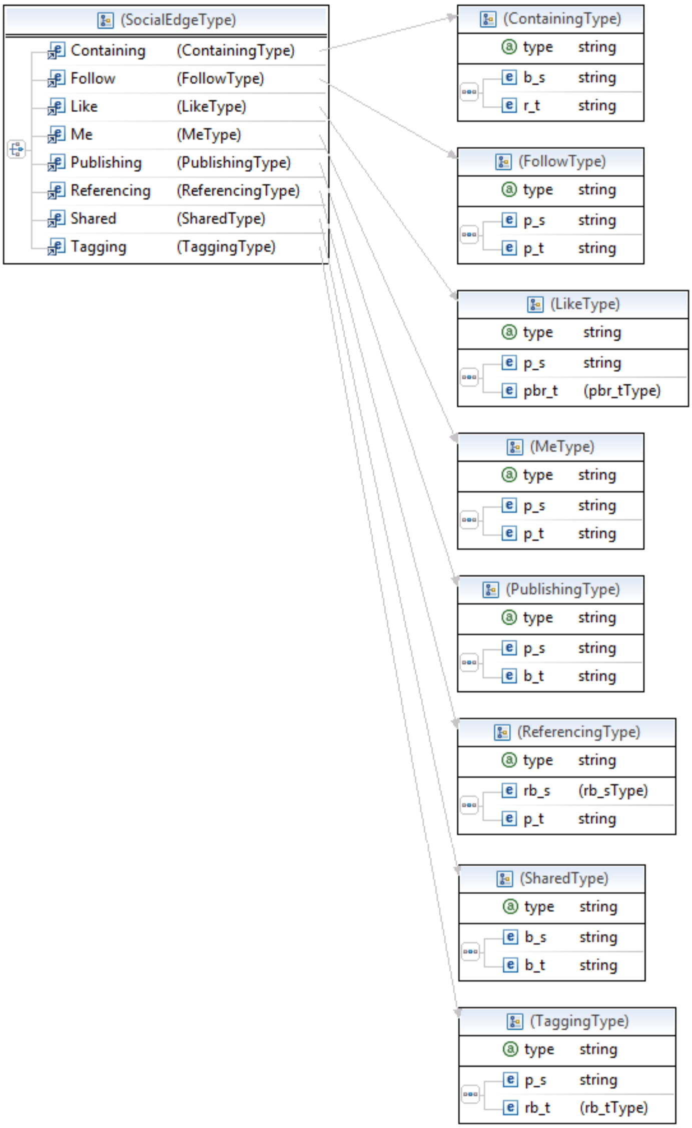

Fig. 3. A portion of the tree representation of our XML Schema (3/3)

The model defined here has been successfully used in the SNAKE system [8], a tool supporting the extraction of data from social network accounts.

```
...
    <element name="SocialGraph">
        <complexType>
            <sequence>
                <element ref="tns:SocialNode" minOccurs="0" maxOccurs="unbounded" />
                <element ref="tns:SocialEdge" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
        </complexType>
    </element>

    <element name="SocialNode">
        <complexType>
            <choice>
                <element ref="tns:SocialProfile" />
                <element ref="tns:Resource" />
                <element ref="tns:Bundle" />
            </choice>
        </complexType>
    </element>

    <element name="SocialProfile">
        <complexType>
            <sequence>
                <element name="URL" type="string" minOccurs="1" maxOccurs="1" />
                <element name="SocialNetwork" type="string" minOccurs="0"
                    maxOccurs="1" />
                <element name="Screen-name" type="string" minOccurs="1"
                    maxOccurs="1" />
                <element name="PersonalInformation" type="string" minOccurs="0"
                    maxOccurs="1" />
                <element name="Picture" type="string" minOccurs="0"
                    maxOccurs="1" />
            </sequence>
        </complexType>
        <xs:key name="spkey">
            <xs:selector xpath="SocialGraph/SocialNode/SocialProfile" />
            <xs:field xpath="@URL" />
        </xs:key>

    </element>
...

    <element name="SocialEdge">
        <complexType>
            <choice>
                <element ref="tns:Containing" />
                <element ref="tns:Follow" />
                <element ref="tns:Like" />
                <element ref="tns:Me" />
                <element ref="tns:Publishing" />
                <element ref="tns:Referencing" />
                <element ref="tns:Shared" />
                <element ref="tns:Tagging" />
            </choice>
        </complexType>
    </element>
...
```

Listing 1.7. A portion of our XML Schema.

The second application that benefits from our model concerns the problem of identifying users on the Web. This problem has received a great attention in several application scenarios, such as personalization [10,14,32].

A common approach to address this problem utilizes profile matching techniques typically based on a set of identification properties, such as username, to find user corresponding identity. In [6], an improvement of this approach is proposed. In particular, a new notion of profile similarity is defined, by combining a string similarity between the associated usernames with a contribution based on a suitable recursive notion of common-neighbor similarity. The computation of the second contribution requires to compare profiles coming from different social networks, which could be quite heterogeneous.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:SocialGraph xmlns:tns="http://www.example.org/SocialGraph"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.example.org/SocialGraph␣SocialGraph.xsd">

     <tns:SocialNode>
            <tns:SocialProfile>
                   <tns:URL>twitter.com/antoninonocera</tns:URL>
                   <tns:SocialNetwork>Twitter</tns:SocialNetwork>
                   <tns:Screen-name>AntoninoNocera</tns:Screen-name>
            </tns:SocialProfile>
     </tns:SocialNode>

     <tns:SocialNode>
            <tns:SocialProfile>
                   <tns:URL>twitter.com/serenanicolazzo</tns:URL>
                   <tns:SocialNetwork>Twitter</tns:SocialNetwork>
                   <tns:Screen-name>serenanicolazzo</tns:Screen-name>
            </tns:SocialProfile>
     </tns:SocialNode>

     <tns:SocialNode>
            <tns:Bundle>
                   <tns:URI>240859602684612608</tns:URI>
                   <tns:Description>Take a look to this beautiful photo :-)</tns:Description>
                   <tns:CreationDate>2012-08-29</tns:CreationDate>
                   </tns:Bundle>
     </tns:SocialNode>

     <tns:SocialNode>
            <tns:Resource>
                   <tns:URI>http://pbs.twimg.com/media/A7EiDWcCYAAZT1D.jpgI</tns:URI>
                   <tns:Type>photo</tns:Type>
                   <tns:Description>A pic of us</tns:Description>
                   <tns:CreationDate>2012-08-29</tns:CreationDate>
            </tns:Resource>
     </tns:SocialNode>

     <tns:SocialEdge>
            <tns:Follow type="F">
                   <tns:p_s>twitter.com/antoninonocera</tns:p_s>
                   <tns:p_t>twitter.com/serenanicolazzo</tns:p_t>
            </tns:Follow>
     </tns:SocialEdge>

     <tns:SocialEdge>
            <tns:Publishing type="Pu">
                   <tns:p_s>twitter.com/antoninonocera</tns:p_s>
                   <tns:b_t>240859602684612608</tns:b_t>
            </tns:Publishing>
     </tns:SocialEdge>

     <tns:SocialEdge>
            <tns:Containing type="Co">
                   <tns:b_s>240859602684612608</tns:b_s>
                   <tns:r_t>http://pbs.twimg.com/media/A7EiDWcCYAAZT1D.jpg</tns:r_t>
            </tns:Containing>
     </tns:SocialEdge>

     <tns:SocialNode>
            <tns:SocialProfile>
                   <tns:URL>www.facebook.com/antonino.nocera.35</tns:URL>
                   <tns:SocialNetwork>Facebook</tns:SocialNetwork>
                   <tns:Screen-name>Antonino Nocera</tns:Screen-name>
                   <tns:PersonalInformation>Male</tns:PersonalInformation>
            </tns:SocialProfile>
     </tns:SocialNode>

     <tns:SocialEdge>
            <tns:Me>
                   <tns:p_s>twitter.com/antoninonocera</tns:p_s>
                   <tns:p_t>www.facebook.com/antonino.nocera.35</tns:p_t>
            </tns:Me>
     </tns:SocialEdge>
     ...
 </tns:SocialGraph>
```

Listing 1.8. An example of an XML document.

It is worth noting that the papers referred above (i.e., [8] and [6]) are completely not focused on the modelling (and implementation) aspects faced in this paper, so that this paper adds original relevant work.

6 Related Work

Over time, social network analysts have mainly used two kinds of tools from mathematics to represent information about patterns of ties among social actors: graphs and matrices.

Examples of the first group are: Kronecker graphs model [19]; the class of model networks presented in [22] that are generalizations of the much-studied random graph of Erdös and Rényi [11] to model social networks; the multiplicative attribute graph model presented in [17] that considers information about properties of the nodes of the network; and others, such as [9,29,28], that model their application scenarios with graphs.

The approaches that adopt matrices representation to model social networks [18,26] belong to the second group. Specifically, the approach of [18] incorporates social influence processes in the specification of a weight matrix W, whereas the approach of [26] uses a tensor to model the interaction between resources and users.

A minor trend is to formalize social networks through a three phase model [24,13]. This model was developed in [30] to identify critical social networks activities.

The hypergraph theory [16] allows a hyperedge to connect an arbitrary number of vertices instead of two in regular graphs. For instance, Ghoshal et al. [12] introduce a random hypergraph model to describe the ternary relationship among one user, one resource and one tag, thus making the model more flexible in the representation of many peculiar properties of folksonomies. In [33], the authors propose an hypergraph model to illustrate the emergence of some statistical properties in a folksonomy such as: degree distribution, clustering coefficients and average distance between nodes.

Another worthy-of-mention category of approaches adopt suitable models with the purpose of creating global user profiles by means of deep analyses of their behavior accessing multiple social networks. Often, the application scenarios of these approaches are those of ontologies and folksonomies.

An interesting approach, in this context, is TAGMAS (Tag MAnagement System) [15]. It relies on an ontology to uniformly describing tags and tagging actions in several distinct folksonomies.

The problem of integrating data of different social sites is addressed in [25] and [23]. The basic idea of [25] is to collect data from different folksonomies and, then, to create groups of tags by adopting suitable clustering algorithms. After this step, these groups are mapped onto the concepts of an ontology. In [23], the authors propose an approach which gathers data about user activities on social sites. Suitable ontologies are used both to analyze these data and model user interests. An important approach belonging to this category is that of [21]. The author formulates an abstract model of semantic-social networks, in the form of a tripartite graph of persons, concepts and instances. Hence, incorporating actors in this model, he extends the traditional concept of ontologies (composed by concepts and instances). By the way, because the referring scenario of this paper is that of folksonomies, this model represents only one action (i.e., *tagging*). It is defined as a ternary association between user, concept and object. More in detail, the set of shared object and the set of keywords defined by users themselves are extracted from social networks. These collections are, then, used to obtain the emergence of a community-based ontologies.

Even if our approach has some common aspects with these proposals, none of them consider the possibility of integrating information coming from different and heterogeneous social networks. This additional feature of our model makes it strongly different from the approaches presented here, because the uniform representation of all the peculiarities of different social networks is a non-trivial task and needs ad-hoc solutions to be pursued.

7 Conclusion

In this paper, we have defined and implemented a model aimed at creating a middleware on top of existing online social networks. The goal is to provide a (conceptual) layer able to facilitate design and implementation of applications relying on the internetworking nature of online social networks. As a matter of fact, the multiplicity of social networks together with users' membership overlap, result in a multiplicative effect in terms of information power. Indeed, correlation, integration, negotiation of information coming from different social networks offer a lot of strategic knowledge whose benefits are still unexplored.

The approach followed in this paper was mainly practical, in the sense that we solved the trade-off between complexity/expressiveness of the conceptual model and implementation issues in favor of the latter. In other words, despite a somewhat naive model focused on a few crucial concepts underlying real-life social networks, the resulting benefits from the implementation perspective appears considerable. This is shown in the paper by means of case study, but we believe that a lot of further applications may confirm the relevance of our approach. We plan to show this in our future work.

Acknowledgment. This work has been partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research and by the Program "Programma Operativo Nazionale Ricerca e Competitività" 2007-2013, project BA2Kno (Business Analytics to Know) PON03PE_00001_1, in "Laboratorio in Rete di Service Innovation", funded by the Italian Ministry of Education, University and Research.

References

1. Videotrine. The most viewed videos on Youtube in the World of all time (2014), http://en.videotrine.com/all/youtube/all-time
2. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.91. Technical report, Tech. rep. ILRT Bristol (2000), http://xmlns.com/foaf/spec/20071002.html
3. Buccafurri, F., Foti, V., Lax, G., Nocera, A., Ursino, D.: Bridge Analysis in a Social Internetworking Scenario. Information Sciences 224, 1–18 (2013)

4. Buccafurri, F., Lax, G., Nicolazzo, S., Nocera, A., Ursino, D.: Driving Global Team Formation in Social Networks to Obtain Diversity. In: Casteleyn, S., Rossi, G., Winckler, M. (eds.) ICWE 2014. LNCS, vol. 8541, pp. 410–419. Springer, Heidelberg (2014)
5. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: Crawling social internetworking systems. In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), pp. 506–510. IEEE Computer Society (2012)
6. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: Discovering Links among Social Networks. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 467–482. Springer, Heidelberg (2012)
7. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: Moving from social networks to social internetworking scenarios: The crawling perspective. Information Sciences 256, 126–137 (2014)
8. Buccafurri, F., Lax, G., Nocera, A., Ursino, D.: A system for extracting structural information from social network accounts. Software: Practice and Experience (2014), doi:10.1002/spe.2280
9. Caldarelli, G.: Scale-Free Networks: Complex Webs in Nature and Technology. Number 9780199211517 in OUP Catalogue. Oxford University Press (2007)
10. Carmagnola, F., Cena, F.: User identification for cross-system personalisation. Information Sciences 179(1-2), 16–32 (2009)
11. Erdös, P., Rényi, A.: On Random Graphs, I. Publicationes Mathematicae 6, 290–297 (1959)
12. Ghoshal, G., Zlatić, V., Caldarelli, G., Newman, M.E.J.: Random hypergraphs and their applications. Physical Review E 79(6), 066118 (2009)
13. Greve, A., Salaff, J.W.: Social networks and entrepreneurship. Entrepreneurship Theory and Practice 28(1), 1–22 (2003)
14. Iofciu, T., Fankhauser, P., Abel, F., Bischoff, K.: Identifying users across social tagging systems. In: Proc. of the International Conference on Weblogs and Social Media (ICWSM 2011), Barcelona, Catalonia, Spain. The AAAI Press (2011)
15. Iturrioz, J., Diaz, O., Arellano, C.: Towards federated web2. 0 sites: The tagmas approach. In: Tagging and Metadata for Social Information Organization Workshop, WWW 2007 (2007)
16. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: applications in VLSI domain. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 7(1), 69–79 (1999)
17. Kim, M., Leskovec, J.: Modeling social networks with node attributes using the multiplicative attribute graph model. arXiv preprint arXiv:1106.5053 (2011)
18. Leenders, R.T.: Modeling social influence through network autocorrelation: constructing the weight matrix. Social Networks 24(1), 21–47 (2002)
19. Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., Ghahramani, Z.: Kronecker graphs: An approach to modeling networks. The Journal of Machine Learning Research 11, 985–1042 (2010)
20. Lovász, L.: Random walks on graphs: A survey. Combinatorics, Paul Erdos is Eighty 2(1), 1–46 (1993)
21. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
22. Newman, M.E.J., Watts, D.J., Strogatz, S.H.: Random graph models of social networks. Proceedings of the National Academy of Sciences 99(suppl. 1), 2566–2572 (2002)

23. Noor, S., Martinez, K.: Using social data as context for making recommendations: an ontology based approach. In: Proceedings of the 1st Workshop on Context, Information and Ontologies, p. 7. ACM (2009)
24. Romm, C., Pliskin, N., Clarke, R.: Virtual communities and society: toward an integrative three phase model. International Journal of Information Management 17(4), 261–270 (1997)
25. Specia, L., Motta, E.: Integrating folksonomies with the semantic web. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)
26. Stewart, A., Diaz-Aviles, E., Nejdl, W., Marinho, L.B., Nanopoulos, A., Schmidt-Thieme, L.: Cross-tagging for personalized open social networking. In: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, pp. 271–278. ACM (2009)
27. Stutzback, D., Rejaie, R., Duffield, N., Sen, S., Willinger, W.: On unbiased sampling for unstructured peer-to-peer networks. In: Proc. of the International Conference on Internet Measurements, Rio De Janeiro, Brasil, pp. 27–40. ACM (2006)
28. Wang, A.H.: Don't follow me: Spam detection in twitter. In: Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT), pp. 1–10. IEEE (2010)
29. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1031–1040. ACM (2011)
30. Wilken, P.H.: Entrepreneurship: A comparative and historical study. Ablex, Norwood (1979)
31. Ye, S., Lang, J., Wu, F.: Crawling online social graphs. In: Proc. of the International Asia-Pacific Web Conference (APWeb 2010), Busan, Korea, pp. 236–242. IEEE (2010)
32. Zafarani, R., Liu, H.: Connecting corresponding identities across communities. In: Proc. of the International Conference on Weblogs and Social Media (ICWSM 2009), San Jose, CA, USA. The AAAI Press (2009)
33. Zhang, Z., Liu, C.: A hypergraph model of social tagging networks. Journal of Statistical Mechanics: Theory and Experiment 2010(10), P10005 (2010)