

A Model for Handling Multiple Social Networks and its Implementation (discussion paper^{*})

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

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

Abstract. Nowadays, users join several on-line social networks (OSNs) so that design and development applications able to work on multiple OSNs is a challenging issue. However, OSNs present relevant differences from both the adopted terminology (similar concepts have different names) and the supported technology (for example, in the APIs provided for data extraction). Consequently, the heterogeneity of OSNs does not allow the design of applications with suitable abstraction with respect to the specific OSNs processed. In this paper, we define a model aimed at generalizing concepts, actions and relationships of existing social networks, which can be exploited as a middleware to implement applications working on multiple social networks.

Keywords Multiple Social Networks, Facebook, Twitter, API.

1 Introduction

Over the past decade, online social networks have become part of people's live. Nowadays, most people have a profile in one or more online social networks like **Facebook**, **Twitter**, **Linkedin**, **MySpace**, in which they spend a lot of time. This is recognized as an important phenomenon from a social and economic point of view, and, thus, in design and development processes of (Web) applications. Indeed, often applications should be based on behaviors of a community, or take advantage from these, so that modern Web applications should be social by default. In many cases, both personal information and social interactions coming from social network profiles can be part of innovative solutions. Among these, social Web applications are the most significant example, in which both people's identities and contents they produced are involved in the business process and data are mostly owned by users, strongly interlinked and inherently polymorphic

^{*} This is a short version of the paper titled "A model to support design and development of multiple-social-network applications" [3], which appears in Information Science journal.

[1]. Indeed, 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: for example, connected people in Facebook are *friends*, whereas they are *followers* or *followings* in Twitter. Consequently, there is the need of delaying the binding between abstract concepts and concrete API calls, when applications operate across multiple social networks: the abstract request of finding connected people is implemented differently in Facebook and Twitter (this argument is discussed in Section 2). This is a strong handicap for the design and implementation of applications enabling internetworking functions among multiple social networks, and, then, for the achievement of the above goal. As a matter of fact, little exists in terms of models and languages to support social-network-based programming in large, according to software engineering principles of genericity and polymorphism.

On the other hand, the power of the social-network substrate can be fully exploited only if we move from a single-social-network to a multiple-social-network perspective, still keeping the user-centered vision, so that the above issue becomes crucial. The recent literature has highlighted that the aforementioned multiple-social-network perspective opens a lot of new problems in terms of analysis [12] but also new opportunities from the application point of view [14, 8, 15, 18]. 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. Thus, social-network-based programming should work at this abstraction level.

In this paper, we do an important step to cover the gap highlighted above, by defining and implementing a model aimed at generalizing concepts, actions and relationships of existing social networks. This paper is organized as follows. Section 2 introduces the characteristics of the multiple-social-network scenario that we model. We give a formal definition of the graph-based conceptual model in Section 3. To validate our approach, in Section 4, we show how our model has been profitably applied to two very relevant applications in the context of social network analysis. Finally, our conclusions are summarized in Section 5.

2 Design specification

One of the motivation of this study is the strong heterogeneity in the representation of concepts among different social networks. For instance, contacts are represented by *friends* in Facebook and the relationship is symmetric, while they are represented by *followers* and *followings* in Twitter and the corresponding relationship is not symmetric. Again, the concept of appreciation becomes *+1* in Google+ and *endorsement* in about.me. Importantly, similar concepts can mapped to each other but they have in general different features. Thus, an integration step is necessary for our purpose. In this section, we prepare this integration step by grouping the main technical entities into a number of categories

to which the formal model presented in the next section maps. In particular, we aim at modeling the following entities.

Profile. Social network sites are built around user profiles, a form of individual (or group) homepage, which provides a description of each registered user. For example, in **Twitter**, at the moment of registration, a user can create his profile typing his name, username, password and email address in the registration form. After, he can upload a profile picture and start following other people. Moreover he can complete his profile adding a short biography, a position (the place where he lives) and a link to his website or to one of his accounts on other social networks. Another social network, **about.me**, is characterized by its one-page user profiles, each with a large background image and short biography. At the moment of registration a user has to fill the suitable form with his username, email, password for the site and at a second step short biography, a short description, a profile image and a background image.

Links to external social networks. An important feature provided by all the social networks considered in this paper is the possibility for a user to add in his profile a link toward one of his accounts in another social site or external website. This feature is typically enabled during the creation of the user profile. It is of particular interest in this paper because it encodes the basic information allowing the possibility of seeing different social sites as members of a Multiple-Social-Network environment.

Friendship. After creating a profile, participants are asked to invite their friends to the site or to look at others' profiles and add those people to their list of friends. In **Twitter**, a user can follow another user, becoming his *follower*. Only if this user follows him back the relationship is bidirectional. Differently from **Twitter**, **Facebook** requires approval for two people to be linked as friends. When someone links another as a friend, the recipient receives a message asking for confirmation. Indeed, **Facebook** friendship is bidirectional, hence, once a user accepts a friendship request of another user they become mutual friends.

Resources. A Social network resource is a Web asset such as a status update, a photo, a web link or a video created and loaded by a user in his profile. As for **LinkedIn**, a user can add a resource like a new item or a new file in his profile. He can also embed a comment, a photo, a web link or a video in a new status update. Also skills representing specific technical expertise can be seen as a typology of resource, which are posted by users to describe their ability. This way, his *connections* can like it, comment it and share it on their "wall".

Actions on resources. So far, we stated that in addition to the content that members add when they create their own profiles, social network sites typically provide the possibility to share resources. After a resource is published by a user, several actions can be performed on this resource: other users can appreciate it, or re-share it, or it can be associated with a user through a mention on his profile. Hereafter, we list the main actions a user can do on a resource according to the different social networks analyzed in this paper.

Once a user write a *tweet* in **Twitter**, it will appear on the homepage of all his followers, who can *reply* to it, make it one of their *favourites* or *retweet* it

(that is, forwarding it again on their own timeline). A *tweet* can contain also a user *mention*. It can be done using the symbol @ followed by the referenced username. To categorize *tweets* by keyword, people use the hashtag symbol # before a relevant keyword or phrase (no spaces) in their *tweets*.

Clicking on *like* option on **LinkedIn** presents some differences w.r.t. the **Facebook** *like* function. Indeed, on **LinkedIn**, when users click on the *like* link underneath the various updates, this immediately forwards that particular update out to all of the user first level *connections*. The *share* option, instead, allows users to either redistribute the article (and partially modify it) as an update to their *connections*, post it to a group (or multiple groups), or forward it in a private message. Similarly to what happens in **Twitter**, also in **LinkedIn** while a user publishes a resource he can mention one of his *connections* with the @ symbol. He can also use a keyword as hashtag using the # symbol.

As for **Flickr**, by clicking on a photostream image, it is possible to open it in the interactive *photopage*, thus allowing users to comment it and to embed it on external websites. Moreover, images can be added to a user favourite list or to user galleries. The main **Google+** page consists of a “stream” of updates, conversations and shared content. A user can make comments underneath content shared by other users, and he can appreciate contents clicking “+1” on it. **Google+** provides the referencing functionality in its posts. A user can mention another user using the + or @ signs.

As for **LiveJournal**, users can interact with resources in different ways. For instance, a user can leave a comment on a post of another user or share it in his blog. He can also add to “Memories” a post. The Memories feature on **LiveJournal** allows the organization of favorite resources with a keyword-based archive system. Thanks to this functionality, a user can also add tags, or descriptive keywords, to his own resources.

All the features of the OSNs described in this section are mapped by our model, and this is formalized in the next section.

3 The conceptual model

To model at an abstract level the entities described in the previous section, 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 the tuple $\langle \text{url}, \text{socialNetwork}, \text{screen-name}, [\text{personalInformation}], [\text{picture}] \rangle$, where `url` is the Web address that identifies and localizes the profile, and `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 $\langle \text{url}, \text{type}, [\text{description}], [\text{date}] \rangle$, 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 who published the resource, describing the resource itself and the publishing date, respectively. For example, the most viewed video on YouTube is a resource represented as $\langle \text{'https://www.youtube.com/watch?v=9bZkp7q19f0'}, \text{'video/mp4'}, \text{'PSY - GANGNAM STYLE'}, \text{'07/15/2012'} \rangle$.

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 $\langle \text{uri}, [\text{description}], [\text{date}] \rangle$, 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. The set E of these edges is partitioned into 8 disjoint sets, named F , M , Pu , S , T , Re , L , and Co .

The *follow* edge set $F \subseteq E = \{p_s, p_t \mid p_s, p_t \in P\}$ models the fact 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 friendships, 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 $Pu \subseteq E = \{p_s, b_t \mid p_s \in P, b_t \in B\}$ indicates 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 $S \subseteq E = \{b_s, b_t \mid b_s, b_t \in B\}$ specifies 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 $T \subseteq E = \{p_s, br_t, w \mid p_s \in P, br_t \in B \cup R \text{ and } w \text{ is a word}\}$, 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 $L \subseteq E = \{p_s, pbr_t \mid p_s \in P, pbr_t \in B \cup R \cup P\}$ describes 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 $Co \subseteq E = \{b_s, r_t \mid b_s \in B, r_t \in R\}$ indicates 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 .

Concerning how to practically map real-life data from social networks to each component of the model, the reader can refer to [3]. In the next section, we show how this model has been exploited at application level.

4 Case studies

Evaluating the accuracy of a model is a difficult task because often a golden standard misses [2]. In these cases, evaluation can be done by humans (e.g., [13, 11]) or by applying the model to an application and evaluating the results (e.g., [16]). In this section, following the latter approach, we describe how our model has been profitably applied to two applications very relevant in the context of social network analysis.

The first application we discuss regards the extraction of information from a multiple-social-network scenario. 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 [19], Random Walk [10] or Metropolis- Hastings Random Walk [17]. 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. In this case, it is possible to exploit a crawling task implementing the following steps.

1. Selecting the starting account (seed). This step is very important to provide data useful to the specified application. Usually, the starting account is randomly selected from an available pool of accounts. For particular analysis, the seed can be selected from those accounts having some characteristics, for example, being a power user (i.e., they have a number of contacts much higher than the average user [9]).
2. Building the sub-graph. In this step, the information about this account is created: it includes the user account, contacts, published resources, and so

on. This step is strongly facilitated by our model. Indeed, by following the procedures described in Section ??, we map all information extracted from the different social networks to the components of our model (i.e., profiles, resources, bundles, and their relationships).

3. Selecting the next account. There exist several strategies to implement this step. A first possibility is to randomly select another profile (uniform sampling), and this is feasible whenever a social network uses an identifier for accounts and the domain of identifiers is known and limited. This occurs for example for Facebook and Twitter [7]. Another possibility consists in selecting one profile (i.e., a node of the graph) connected with the last visited profile by a *follow* edge or a *me* edge (see, for example, [10, 17]). Again, it is also possible to select more than one (even all) of the profiles referred above, as done for example in [4, 19]. Once one or more profiles have been selected, Steps 2 and 3 are iterated until the desired amount of data have been extracted or a stop condition has been reached.

The model defined here has been successfully used in the SNAKE system [6], a tool supporting the extraction of data from social network accounts.

The second application that benefited from our model concerns the problem of identifying users on the Web. 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 [5], 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. The use of our model allowed us to simplify this issue and to handle all profiles in a uniform way. We can state that the success of the technique described in [5] strongly relied on the model described in this paper.

5 Conclusion

It is a matter of fact that 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. 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. By means of two case studies, we showed the effectiveness of the proposed model.

References

1. G. Bell. *Building social web applications*. "O'Reilly Media, Inc.", 2009.

2. J. Brank, M. Grobelnik, and D. Mladenić. A survey of ontology evaluation techniques. In *In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, 2005.
3. F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. A model to support design and development of multiple-social-network applications. *Information Sciences*, 331:99–119, 2016.
4. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Moving from social networks to social internetworking scenarios: The crawling perspective. *Information Sciences*, 256:126–137, 2014. Elsevier.
5. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Discovering missing me edges across social networks. *Information Sciences*, 319:18–37, 2015.
6. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. A system for extracting structural information from social network accounts. *Software: Practice and Experience*, 2015. DOI: 10.1002/spe.2280.
7. M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *Proc. of the International Conference on Computer Communications (INFOCOM'10)*, pages 1–9, San Diego, CA, USA, 2010. IEEE.
8. M. N. Jelassi, C. Largeton, and S. B. Yahia. Efficient unveiling of multi-members in a social network. *Journal of Systems and Software*, 94:30–38, 2014.
9. S.-H. Lim, S.-W. Kim, S. Park, and J. H. Lee. Determining content power users in a blog network: an approach and its applications. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(5):853–862, 2011.
10. L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2(1):1–46, 1993.
11. A. Lozano-Tello and A. Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 2(15):1–18, 2004.
12. V. S. A. Menezes, G. Zimbrão, and J. M. Souza. Group and link analysis of multi-relational scientific social networks. *Journal of Systems and Software*, 86(7):1819–1830, 2013.
13. P. Mika. Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web–ISWC 2005*, pages 522–536. Springer, 2005.
14. D. T. Nguyen, H. Zhang, S. Das, M. T. Thai, and T. N. Dinh. Least cost influence in multiplex social networks: Model representation and analysis. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 567–576. IEEE, 2013.
15. A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos. Fast and accurate link prediction in social networking systems. *Journal of Systems and Software*, 85(9):2119–2132, 2012.
16. R. Porzel and R. Malaka. A task-based approach for ontology evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*, 2004.
17. D. Stutzback, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. On unbiased sampling for unstructured peer-to-peer networks. In *Proc. of the International Conference on Internet Measurements*, pages 27–40, Rio De Janeiro, Brasil, 2006. ACM.
18. Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang, and H. Yan. Recommender systems based on social networks. *Journal of Systems and Software*, 99:109–119, 2015.
19. S. Ye, J. Lang, and F. Wu. Crawling online social graphs. In *Proc. of the International Asia-Pacific Web Conference (APWeb'10)*, pages 236–242, Busan, Korea, 2010. IEEE.