Social Networking Interoperability through Extended FOAF Vocabulary and Service

Bo Zhou
Department of Computer Science
Zhejiang University
Hangzhou, P.R. China
bzhou@zju.edu.cn

Chao Wu
Department of Computer Science
Zhejiang University
Hangzhou, P.R. China
wuchao@zju.edu.cn

Abstract— **Various social network services (SNSes) cause the problem of social data inconsistency. The key to resolve this problem is to bring interoperability among multiple social network services. In this paper, we propose to achieve this by introducing the ontology representing SNSes based on FOAF vocabulary. We also implement a FOAF web service to act as an intermediary for social networking interoperability. The example application shows its usage and advantages over the conventional approach.**

Keyword SNS; FOAF; social networking interoperability

I. INTRODUCTION

Current social networking is mainly characterized by a large set of diverse social network services. A typical user has multiple online social networks from SNSes, such as Facebook[1], LinkedIn[2], Twitter[3], etc, as shown in Figure 1. Notice the overlap exists between different SNSes. Some contacts and their information in a network (e.g. LinkedIn) may simultaneously exist in another network (e.g. Twitter) of the same user.

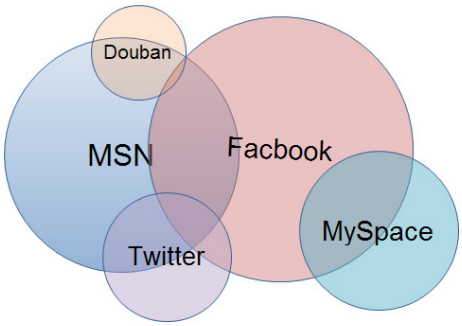

Figure 1. Multiple online social networks for a user

Users usually have multiple SNSes simultaneously. The data distribute among various sites, is inconsistent and difficult for interoperability:

a) Ideally, a user's social network should be unique and uniform, rather than scatted across different social network services. With current approach, for example, it's impossible to chat with a person in your blogsphere network if he is not in your instant message network;

b) Data redundancy makes information on personal social contacts hard to maintain. When you want to modify contact information of a person, you need to replay the action in several places.

c) Data in each site might be incomplete and difficult to merge. You might find one's email in site A, and his telephone number in site B.

d) The data are controlled by social network operators, and difficult for migration. When you turn to a new social network service, it's hard to establish a new social network from the scratch.

All these issues and limitations in current social networking motivated our work to provide a framework for supporting interoperability of social networking systems.

We will firstly define a ontology model to represent social network services, which provide the basis for different SNSes to communicate. We then construct a web service server for social ontology with the function of service interoperation. With this approach, users and various SNSes could manipulate the social networking data in a uniform way. Social networking applications would be more powerful with interoperability.

II. RELATED WORK

Although the concept of social networking could date back to the 1960s, viral growth and commercial interest only arose well after the advent of the Internet. According to [1], the first group of recognizable social networking sites launched since 1997 after the dramatic rise of Friendster[4]. In recent years, online social networking has moved from a niche phenomenon to mass adoption. The popularity of social network services like Facebook, Flickr[5], and Del.icio.us[6] has drawn great attention from the research community [4, 14, 7, 12, 3, 6].

[1] http://www.facebook.com
[2] http://www.linkedin.com
[33] http://www.twitter.com

[4] http://www.friendster.com
[5] http://www.flickr.com
[6] http://del.icio.us

Some efforts have been made to introduce semantics into social networks. For example, Mika [8] used semantic web methods to extract and analyze the social networks of research community. Parunak [4] tried to represent the social connects with a UML-like way. The FOAF project is a first attempt towards developing a formal, machine processable representation of user profiles and friendship networks. Paolillo and Wright offered a rough characterization of the FOAF web in [11]. There have already been applications of FOAF network, such as expert finding [15] and trust network constructing [13]. Kruk [17] used FOAF to control the resource access by evaluating the friendship between people. Mori [16] proposed a method to automatically produce keyword for building FOAF document. Li [5] gave an analysis of current usage of FOAF document and network. There are also many tools and applications supporting FOAF [9, 10].

From the industry side, there are some efforts trying to provide uniform paradigm for social networking. The most remarkable jobs include Facebook platform [7] and Google OpenSocial[8] architecture. However, both of them have obvious limitation. Facebook platform provides developers a framework to create applications interacting with Facebook social networking. It has visible advantage due to the popularity of the system. With its API, applications could utilize the existing social networking, and the data is consistence across multiple applications. If the user has only Facebook for social networking on the Internet, this solution is promising. However, as we stated, for each individual, usually there are many other SNSes in use, and Facebook's applications could not easily access those ones. OpenSocial is a set of API developed by Google for web-based social network applications. Applications implementing the OpenSocial APIs will be interoperable with any social network system that supports them. The apps could be "coded once, used everywhere". However, it is actually "Open Widget", the problem of data dispersedness and lack of semantic remains.

In addition, many efforts have been made on SNS mash-up, especially with the web services or data in RSS provided by Web 2.0. These applications mainly focus on visualizing some data by combining another source of data. However, due to the lack of definition of ontology and interoperability among SNSes, mash-up work is inflexible: For example, a developer can use the network from the site A and services from the site B to make a mash-up application, utilizing their open APIs. However, next time when a similar function is demanded to mashup network from A and services from another site C, the developer needs to get familiar with APIs from site C and rewrite many codes – In such a approach of developing social networking application, developers should be familiar with every operation in multiple SNSes, which is really costly in terms of both time and skills. It is also important to note that mash-up technology pay little attention of a uniform querying mechanism to the mashed up social networking systems. For example, a new data source from a SNS requires new code to be written, and changes to the underlying data source may easily break the application.

[7] http://developer.facebook.com
[8] http://www.opensocial.org

III. PREPARE YOUR PAPER BEFORE STYLING

Our approach is to develop such a semantic model by extending the FOAF vocabulary to support SNSes.

A. FOAF Ontology

The FOAF project is one of the first attempts at a formal, machine executable representation of social networking. It defines an RDF vocabulary for expressing metadata about people, and their interests, relationships and activities. Below is an example of FOAF description of a person. FOAF uses "foaf:Person" to define a person, with his/her name ("foaf:name"), email ("foaf:mbox"), etc. The link between persons is defined by "foaf:knows".

```
<foaf:Person rdf:about="http://g.org/cw"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
<foaf:name>Wu Chao</foaf:name>
<foaf:mbox rdf:resource="mailto:cw@fg.org"/>
<foaf:homepage rdf:resource="http://fg.org"/>
<foaf:img rdf:resource="/images/me.jpg" />
<foaf:knows>
<foaf:Person>
<foaf:mbox
   rdf:resource="mailto:bzhou@zju.edu.cn" />
<foaf:name>Bo Zhou</foaf:name>
</foaf:Person>
</foaf:knows>
```

This brief example basically says, "there is a foaf:Person with a foaf:name property of 'Wu Chao' and a foaf:mbox property of mailto:cw@fg.org; this person stands in a foaf:homepage relationship to a thing called http://fg.org and a foaf:img relationship to a thing referenced by a relative URI of /images/me.jpg, this person knows another foaf:Peson with the foaf:name property of 'Bo Zhou' with a foaf:mbox property of mailto: bzhou@zju.edu.cn".

Such FOAF documents are created and managed by individual users in a distributed fashion. They are typically posted on the personal website of the user and linked from the user's homepage or blog with the HTML META tag. The open, semantic, and decentralized feature of FOAF is very suitable to define social networks with dynamical evolution: a) the FOAF social networks are distributed. It allows users to create and store their FOAF descriptions by themselves, and thus avoids the problem of single point of failure, like the denial of service attack to Twitter in August 2009[9]; b) the FOAF networks are open. Personal information is maintained by users themselves, which avoids the abusing of personal information by operators. And due to the open nature of RDF, users can freely add their

[9] http://money.cnn.com/2009/08/06/technology/twitter_outage/index.htm

vocabulary to FOAF for different purposes; c) the FOAF social networks are semantic. The ontology model of the FOAF supports the reasoning over the descriptions.

However, we think a problem hinders the FOAF from being widely adopted to build up online social networks: FOAF is not application oriented, and separated from popular SNSes. On the Internet, people construct their social networks through using SNSes, such as adding contacts in MSN messenger, and making friends in an online game. Users' social networks are always associated with these applications, rather than through manually defining FOAF documents.

B. SNS Extension

The main usage of online social network by end users is using various SNSes, most of which are web based applications. A typical user usually has multiple social network services on the Internet, each containing and controlling a particular social network. The social network data in these social network services are accumulated by users' daily usage. For example, in Del.icio.us, a user might find a friend with similar interest while searching for bookmarks of some topic.

It's difficult for users to discard their social network services and turn to a totally new semantic social network, since both the contacts and contents of them in social network services are important for them. That's why a semantic and uniform social network is difficult to setup. And we also think it's unnecessary to build a new semantic and uniform social network from the scratch. It's more reasonable to combine existing social network services based on the idea of unifying their semantics. We proposed to bridge the gap between these social network services and FOAF ontology by introducing social network service into FOAF framework. Through this approach, we not just connect the FOAF to social network services, but also connect multiple social network services with each other.

In our semantic model, we will present users' profiles and contacts in social network services in FOAF. Besides, most popular social network services provide web services (SOAP, Restful, etc) as interfaces for users and applications to gain data and do the operations. These services need be also represented (with the support for authentication method such as OAuth).

As a result, we extend the FOAF with new properties, as shown in Table 1. The namespace is at:

http://federation-gateway.org/ssra/sns/.

TABLE I. PROPERTIES FOR SOCIAL NETWORK SERVICE

Property	Definition
service	A social network service.
username	User identifier in the social network service.
provider	Social network service's provider.
expires	The expiration time for the information to update.
url	The url of the social network service.
openid	The OpenId of user for this social network service.
profile	User profile in social network services.
contact	User's contact in social network services
priority	The priority of a social network service.
rs	Restful services provided by social network services.
ws	Web services provided by social network services.
op_name	The name of operation.
op_url	The url of operation.
op_se	The authentication method of operation.
response	The response of operation.
argument	Argument for operation.
arg_name	The name of argument.
arg_description	The description of argument.

Below example is a FOAF fragment describing some social network services for a user, including its name, url, priority, etc., as well as a restful operation provided by this social network service.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
      syntax-ns#" xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:st="http://federation-gateway.org/ssra/sns/">
 <sns:service rdf:ID="s00000001">
<dc:about>Delicious</dc:about>
<sns:url>http://delicious.com/mike</sns:url>
<sns:username>Mike</sns:username>
<sns:provider>http://delicious.com</sns:provider>
<sns:priority>10</sns:priority>
<sns:expires>2009-05-5T00:15:000Z</sns:expires>
<sns:profile>http://delicious.com/mike</sns:profile>
<sns:contact>
http://delicious.com/mike/network
</sns:contact>
<sns:rs>
<sns:op_name>addPost</sns:op_name>
<sns:op_url>
https://api.del.icio.us/v1/posts/add
</sns:op_url>
<sns:op_se>https</sns:op_se>
<sns:op_description>
Add a bookmark
</sns:op_description>
<sns:argument>
        <sns:arg_name>url</sns:arg_name>
```

<sns:arg_description>
The url for the bookmark
</sns:arg_description>
</sns:argument>
<sns:argument>
<sns:arg_name>tag</sns:arg_name>
<sns:arg_description>
The tags for the bookmark
</sns:arg_description>
</sns:argument>
</sns:rs>
</sns:service>

Now we've integrated the SNS and FOAF ontology. Each SNS is defined by a <sns:service> tag. SNSes such as Twitter and LinkedIn can find each other in <sns:service>, get the details of counterparts, and communicate with each other through listed operations.

C. FOAF Web Service

We propose to represent the FOAF document as web service. Once there is a request, the web service will produce a fragment of FOAF containing appropriate content for response.Users, social network services, and other social networking applications could gain the machine-readable data (user profile, networks, operations, etc.) from this service.

The FOAF service now supports the interoperation among multiple social network services. SNSes are linked together into a unified system. Social network services could exchange data with each other, and register their operations, for other social network services to call. When an operation is called, the information such as URL, password for security access is protected from the caller. By this way, a social network service could utilize the data from other social network service. For example, since both the SNSes and their operations are represented in <sns:service> in ontology, users of Twitter could send Twitter message to their friends in LinkedIn, just like what we will do in the next section.

IV. Example Application

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

To demonstrate the application of our model, we also developed a mash-up web. The application, called "Linked.Twitter.In", links two popular SNSes: LinkedIn and Twitter. The user could combine the network from his LinkedIn with the message function from Twitter. So it's possible to send Twitter message to LinkedIn contacts.

All the actions of retrieving data and calling operations are done through FOAF web service. The operation for retrieving LinkedIn network is represented as below and stored in FOAF document. Notice that this operation needs the OAuth authentication.

<sns:service rdf:ID="s00000086">
<dc:about>LinkedIn</dc:about>
<sns:url>http://cn.linkedin.com/pub/chao-wu/11/4a0/683</sns:url>
<sns:username>Chao Wu</sns:username>
<sns:provider>http://www.linkedin.com</sns:provider>
<sns:priority>15</sns:priority>
……
<sns:rs>
<sns:op_name>connections</sns:op_name>
<sns:op_url>
http://api.linkedin.com/v1/people/~/connections
</sns:op_url>
<sns:op_se>OAuth</sns:op_se>
<dc:description>
Get network of a person
</dc:description>
<sns:response>XML</sns:response>
</sns:rs>
……
</sns:service>

And the operation for send direct message in Twitter is shown as below:

<sns:service rdf:ID="s00000043">
<dc:about>Twitter</dc:about>
<sns:url>
http://www.twitter.com/wuchao
</sns:url>
<sns:username>idril</sns:username>
<sns:provider>
http://www.twitter.com
</sns:provider>
<sns:priority>15</sns:priority>
<sns:rs>
<sns:op_name>

direct_messages new

</sns:op_name>

<sns:op_url>

http://twitter.com/direct_messages/new.json

</sns:op_url>

<sns:op_se>OAuth</sns:op_se>

<dc:description>

Sends a new direct message to the specified user from the authenticating user. Requires

both the user and text parameters. Request must be a POST. Returns the sent message in the requested format when successful.

</dc:description>

<sns:response>XML</sns:response>

<sns:argument>

<sns:arg_name>user</sns:arg_name>

<sns:arg_description>

Required. The ID or screen name of the recipient user.

</sns:arg_description>

</sns:argument>

……

</sns:rs>

</sns:service>

With these two operations we could get a user's LinkedIn network and send message to his friends in Twitter. All these functions are provided by FOAF web service, the application needs not to deal with the actual SNSes.

In using Linked.Twitter.In, a user should firstly provide his identification. Then the application would redirect the web page to LinkedIn for OAuth authentication:

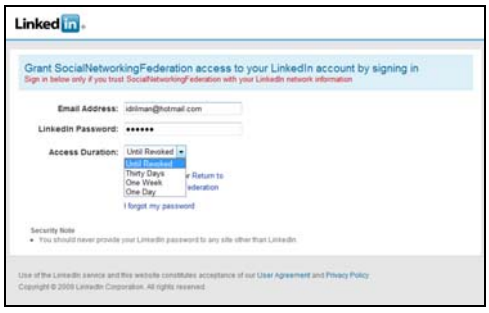

Figure 2. OAuth authentication by LinkedIn

Once the user allows the Linked.Twitter.In to access his network, all his contacts in LinkedIn would be retrieved and listed:

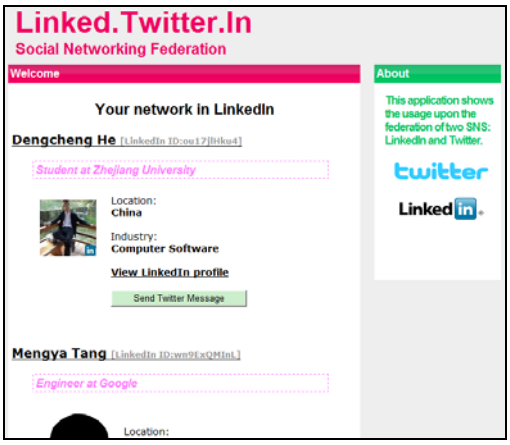

Figure 3. User's network in LinkedIn

Now the user could send Twitter message directly to his connections:

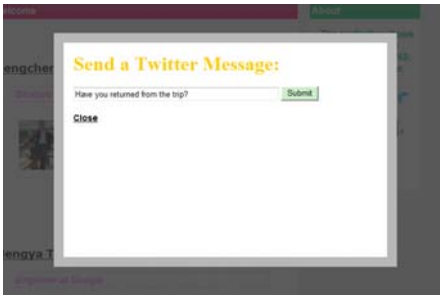

Figure 4. Send Twitter message

We've demonstrated an example system of our approach. Although the example is simple, it shows a simple way to get access to multiple SNSes, integrate a variety of information, and achieve the data / operation federation.

V. CONCLUSION

In this paper, we propose to represent various SNSes and their operations in semantic way by extending the FOAF vocabulary. And then achieve the SNS interoperability through FOAF web service. We believe this work would provide a fundamental part of future social networking development.

Much work remains. First, other than SNS, we need to model some other aspects of social networking in semantic way to support further interoperability. Second, we need to make further design of FOAF web service for better performance on the scale of the Internet. Third, to achieve the power of our approach, we would like to get more support from the operators of SNS.

REFERENCES

[1] boyd, d. m. Ellison, N. B., Social Network Sites: Definition, History, and Scholarship, Journal of Computer Mediated Communication, 2007, 13(1): p. 210-230.

[2] Van Dyke Parunak, H. and J. Odell, Representing Social Structures in UML, in Agent-Oriented Software Engineering II. 2002. p. 1-16.

[3] Brown, G., et al., Social Networks and Context-Aware Spam. Cscw: 2008 Acm Conference on Computer Supported Cooperative Work, Conference Proceedings, 2008: p. 403-412.

[4] Specia, L. and E. Motta, Integrating folksonomies with the semantic web. Semantic Web: Research and Applications, Proceedings, 2007. 4519: p. 624-639.

[5] Ding, L., et al., How the Semantic Web is Being Used: An Analysis of FOAF Documents, in Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04. 2005, IEEE Computer Society.

[6] Du, B. and E.A. Brewer, Dtwiki: a disconnection and intermittency tolerant wiki, in Proceeding of the 17th international conference on World Wide Web. 2008, ACM: Beijing, China.

[7] Dubinko, M., et al., Visualizing Tags over Time. Acm Transactions on the Web, 2007. 1(2).

[8] Mika, P., Flink: Semantic Web technology for the extraction and analysis of social networks. Web Semantics: Science, Services and Agents on the World Wide Web, 2005. 3(2-3): p. 211-223.

[9] FoafNet. Available from: http://foaf.qdos.com/.

[10] FOAF Builder. Available from: http://foafbuilder.qdos.com/.

[11] John C. Paolillo and Elijah Wright. The Challenges of FOAF Characterization. In Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web, 2004.

[12] Gaonkar, S., et al., Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation. Mobisys'08: Proceedings of the Sixth International Conference on Mobile Systems, Applications, and Services, 2008: p. 174-186.

[13] Golbeck, J., B. Parsia, and J. Hendler, Trust Networks on the Semantic Web, in Cooperative Information Agents VII. 2003. p. 238-249.

[14] Goerlitz, O., S. Sizov, and S. Staab, Tagging-based distributed content sharing. Semantic Web: Research and Applications, Proceedings, 2008. 5021: p. 807-811.

[15] J. Li, H. Boley, V.C. Bhavsar, and J. Mei. Expert Finding for eCollaboration Using FOAF with RuleML Rules. 2006 Conference on eTechnologies. Montreal, Canada, 2006.

[16] J. Mori, Y. Matsuo, Mitsuru Ishizuka, and Boi Faltings. Keyword Extraction from the Web for FOAF Metadata. In Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web, 2004.

[17] S. R. Kruk. Foaf-realm - control your friends' access to the resource. In FOAF Workshop proceedings, http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/foaf realm/, 2004.