From Causal History to Social Network in Distributed Social Semantic Software

Khaled Aslan, Hala Skaf-Molli, Pascal Molli

From Causal History to Social Network in Distributed Social Semantic Software

Khaled Aslan
Université de Lorraine, UHP
LORIA, France
aslanalm@loria.fr

Hala Skaf-Molli
Université de Lorraine, UHP
LORIA, France
skaf@loria.fr

Pascal Molli
Université de Lorraine, UHP
LORIA, France
molli@loria.fr

ABSTRACT

Web 2.0 raises the importance of collaboration powered by social software. Social software clearly illustrated how it is possible to convert a community of strangers into a community of collaborators producing all together valuable content. However, collaboration is currently supported by collaboration providers such as Google, Yahoo, etc. following "Collaboration as a Service (CaaS)" approach. This approach arises privacy and censorship issues. Users have to trust CaaS providers for both security of hosted data and usage of collected data.

Alternative approaches including private peer-to-peer networks, friend-to-friend networks, distributed version control systems, distributed peer-to-peer groupware, support collaboration without requiring a collaboration provider. Collaboration is powered with the resources provided by the users.

If it is easy for a collaboration provider to extract the complete social network graph from the observed interactions. Obtaining social network informations in the distributed approach is more challenging. In fact, the distributed approach is designed to protect privacy of users and thus makes extracting the whole social network difficult.

In this paper, we show how it is possible to compute a local view of the social network on each site in a distributed collaborative system approach.

Keywords

Distributed Collaboration System, Semantic Wiki, Causal History, Social Network

1. INTRODUCTION

Software as a service is the current trend in software deployment where an application is hosted as a service and provided to users across the Internet. This model inspired the *collaboration as a service*, where a service provider offers collaborative and social network services.

The social relations provided by the social services are important to push further the collaboration between people. Since it is important to evaluate the location of actors in the network in order to understand networks and their participants; measuring the network location is essential. These measures give us insight into the various roles and groupings in a network: where are the clusters and who is in them, who is in the core of the network, and who is on the periphery.

The social service provider has access to all the data which arises privacy and censorship issues [7], since it can exploit the whole social network relations and interactions among the users.

In order to overcome these issues, new decentralized approaches were proposed. They provide collaborative services without a dedicated service provider. Users can create their own collaborative network and share the collaborative services offered by the system using their own resources. Skype [9] and Distributed Version Control Systems (DVCS) [1] (e.g. Git [8] and Mercurial [13]) demonstrated that it is possible to communicate and share data without the need for a collaboration provider. Although distributed systems provide the required collaborative services, they do not provide the social services offered by centralized systems. In distributed systems there is no central point with a global vision of the social network which is able to build and reflect the social relations among people.

Other approaches use private *peer-to-peer* networks [17] where the resources and the infrastructure are provided by the users participating in the network. Groove [14] is a groupware for collaborative editing which consists of isolated local networks. This system provides group-based network service which allows direct connections between the users of the group. Multi-synchronous semantic wiki [16] is another approach which allows direct connections between sites who know one another i.e. *friend-to-friend* network [4]. However, the collaboration model is very different from CaaS approach. With CaaS software, users mainly collaborate through read-write operations in one shared space provided by the collaboration provider. While in distributed social software, collaboration occurs by replicating shared data and synchronizing multiple workspaces continuously.

The previous model hides the social relations among sites participating in the network. Synchronizing workspace requires exchanging causal histories of operations. By analyzing the causal history on each site, we are able to reveal the social relations and reconstruct on each site a social network graph. So we can see our friends of friends.

This graph represents a local view of the social network and not the whole social network. Users can see their locations in their own local graphs. This approach can enrich the collaboration among the sites with new social services, while at the same time preserving the sites' privacy, since every site has a local vision of the network, and they do not

Web Science Conf. 2010, April 26-27, 2010, Raleigh, NC, USA.
.

rely on a service provider to maintain the social network.

In this paper we extend the collaboration services provided by multi-synchronous semantic wiki with new social network services. We infer the social relations using semantic reasoning by exploiting the semantic causal history. It allows users to discover their friend-of-friend relations. However it does not allow them to connect to their friends of friends. In Section2 we present the multi-synchronous semantic wiki system. Then in Section3 we explain our approach in details. In Section4 we validate the approach and discuss the obtained results. Finally we conclude the paper in Section5.

2. MULTI-SYNCHRONOUS SEMANTIC WIKI

Semantic wikis have emerged as a new generation of collaborative editing tools, we have many examples such as Semantic MediaWiki [10], IkeWiki [18], SWooki [21] c'estand SweetWiki [5]. They support mass collaboration for editing structured and unstructured data. In standard collaborative applications, when a modification is made by one user, it is immediately visible by others. However, in multi-synchronous applications, modifications made by one user is not visible by others. It becomes visible only when a user validates his modifications (commits his changes). A visible change does not imply immediate integration by others. Concurrent modifications will be integrated only when users will decide it.

In a multi-synchronous semantic wiki approach [16], users are allowed to build their own cooperation network by explicitly declaring with whom they would like to cooperate. Every user can run a multi-synchronous semantic wiki server on her machine, and she can create and edit semantic wiki pages locally, then she can share these pages with others. Moreover she can decide with whom to share these pages, and from whom to accept modifications. By this way users create a friend-to-friend network.

In fact sharing a modification in multi-synchronous semantic wiki is accomplished using a capability-based access control approach [12]. A user who modified a page can push a capability to users with whom she would like to share these modifications. This will permit the selected users to integrate the modifications into their local copies; if they decide to pull these modifications.

The modifications are stored in *patches*. We have a *previous* relation defined between two patches to guarantee the causality. So the set of patches constitutes the causal history which is necessary to satisfy the CCI criteria [22]. CCI consistency means Causality, Convergence, and Intention preservation. Thus, causality criterion ensures that all operations ordered by a precedence relation, in the sense of the Lamports happened-before relation [11] will be executed in same order on every site. The system converges if all replicas are identical when the system is idle (eventual consistency). Intention preservation means that an operation effect observed on a copy, must be observed in all copies whatever any sequence of concurrent operations applied before. Actually there is also a changeset concept defined in multi-synchronous semantic wiki; which contains the patches, but we omitted the discussion of this concept for simplicity.

The collaboration model in a multi-synchronous semantic wiki hides the social relations among the sites. We will reveal these relations by transforming the *previous* relation between the patches into social relations among the sites

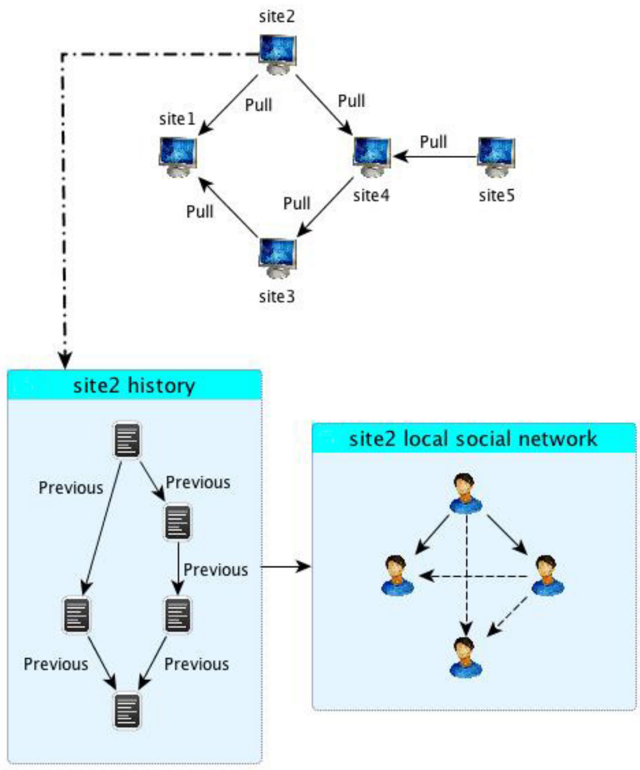


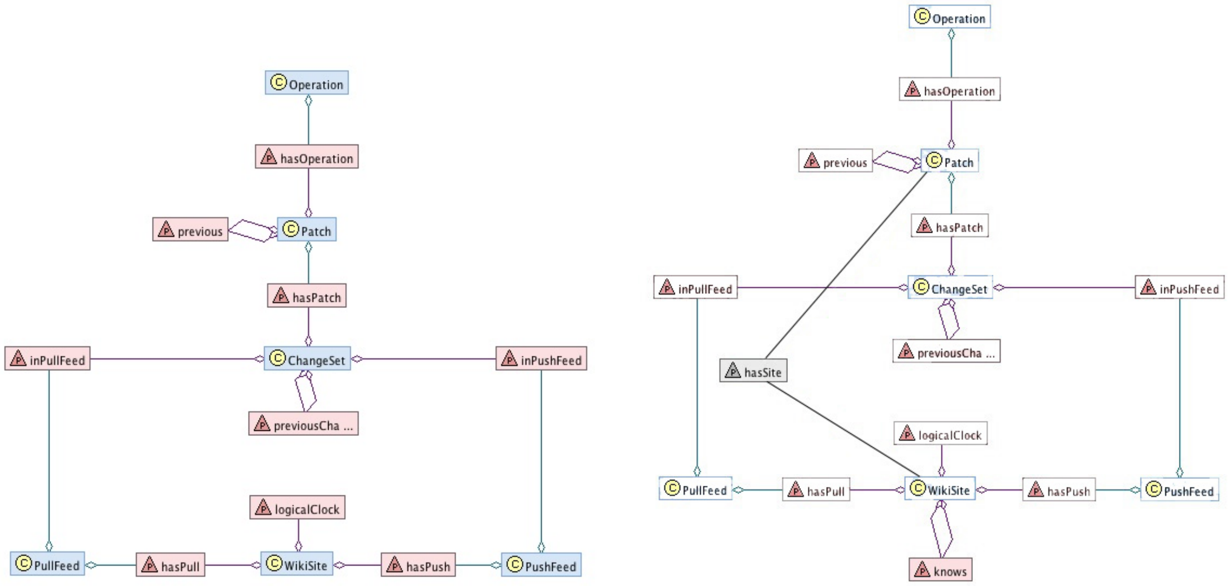
Figure 1: General approach illustration

participating in this network. In figure 1 shows an illustration of our approach. We show how we are going to discover the friend-of-friend relations. In this example we see the interaction between the sites participating in the network. It should be noted that in reality this interaction is not known by any site. This interaction generated a causal history on *site*2 for instance. This site has pulled from two sites only (*site*1 and *site*4) so it has a direct friend relation with these two sites. But it does not know the existence of the other sites in the network. By investigating its own causal history *site*2 will find patches generated on *site*3 and these patches has been pulled by *site*4. So now *site*2 knows the existence of *site*3 and the existence of a relation between *site*4 and *site*3. The deduced knowledge is represented by the dotted arrows in figure 1, this knowledge does not apply that *site*2 can pull from *site*3 since it does not have the capability required to pull from it.

The next section presents how we can reconstruct this social network among the sites, visualize those relations and furthermore calculate the network centrality measures.

3. FROM CAUSAL HISTORY TO SOCIAL RELATIONS

Although there is no direct friendship relation defined in a multi-synchronous semantic wiki, users can manage their relations with others implicitly by controlling from whom to accept modifications and to whom send or publish their modifications. This interaction is recorded in the causal history which is stored at each user's site. Each site participating in a multi-synchronous semantic wiki network keeps



(a) DSMW ontology snapshot (b) DSMW ontology extension snapshot

Figure 2: DSMW ontology

a complete causal history of all the operations it has received or generated locally. The only implementation of a multi-synchronous semantic wiki to the best of our knowledge is DSMW [20]. DSMW is an extension to Semantic-MediaWiki [10]. A snapshot of DSMW ontology is shown in figure 2(b).The figure shows the entities that are of interest for our work only.

According to DSMW ontology, a *patch* is a collection of operations generated at one site, and it is linked to other *patches* by the *previous* transitive relation. Each patch contains the following information:

-PatchID: which is a combination of the siteID and the site logical clock.

-onPage: the page where the patch was applied.

-hasOperation: pointer to the operations generated during the save of the page.

-previous: pointer to the precedent patch.

In order to be able to build the social relation among the sites, we extend DSMW ontology as follows:

1- We define *hasSite* object property from Patch to WikiSite as follows:

$$hasSite \equiv \exists(hasPull^{-1}).\exists(inPullFeed).\exists(hasPatch^{-1}).Patch$$

2- We add an object property *knows* which check if one site knows another site. We check if we have a *previous* relation between patch $P1$ generated on site $S1$ and patch $P2$ generated on site $S2$, if that is the case then this means that $S1$ pulled $P2$ from $S2$ which eventually means that $S1$ *knows* $S2$. This is calculated using the following inference rule:

$$knows(S1, S2) \sqsubseteq \exists P1.(hasSite(P1, S1) \sqcap \exists P2.(hasSite(P2, S2) \sqcap previous(P2, P1)$$

Where $P1, P2$ are both patches, and $S1, S2$ are both sites.

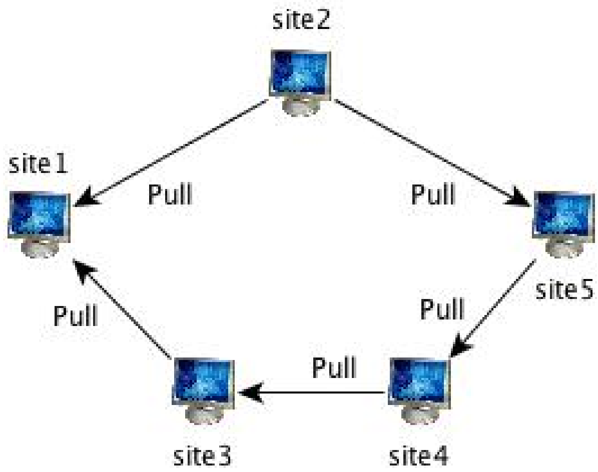


Figure 3: Collaboration scenario

By taking the previous modifications into account we can rebuild a "FOAF:knows" relation [15] between the sites based on the push/pull feeds with a simple inference from the history using a semantic reasoner. Figure 2(b) shows these modifications.

The example in figure 3 shows the interaction between five sites. *site*1 creates a wiki page, then *site*2 and *site*3 pull form *site*1; eventually *site*2 modifies the page; at the same time *site*3 makes some modifications too, then *site*4 pulls from *site*3; by its turn *site*4 modifies the page, then *site*5 pulls from *site*4 and modifies the page too. Finally *site*2 pulls from *site*5.

By investigating *site*2's history shown in figure 4(a). We

(a) site2's causal history

```
MS2W:Patch1
      MS2W:hasSite MS2W:Site1 ;
      MS2W:onPage MS2W:lesson1 .
MS2W:Patch2
      MS2W:hasSite MS2W:Site2 ;
      MS2W:onPage MS2W:lesson1 ;
      MS2W:previous MS2W:Patch1 .
MS2W:Patch3
      MS2W:hasSite MS2W:Site3 ;
      MS2W:onPage MS2W:lesson1 ;
      MS2W:previous MS2W:Patch1 .
MS2W:Patch4
      MS2W:hasSite MS2W:Site4 ;
      MS2W:onPage MS2W:lesson1 ;
      MS2W:previous MS2W:Patch3 .
MS2W:Patch5
      MS2W:hasSite MS2W:Site5 ;
      MS2W:onPage MS2W:lesson1 ;
      MS2W:previous MS2W:Patch4 .
MS2W:Patch6
      MS2W:hasSite MS2W:Site2 ;
      MS2W:onPage MS2W:lesson1 ;
      MS2W:previous MS2W:Patch5 .
```

(b) site2's local social network

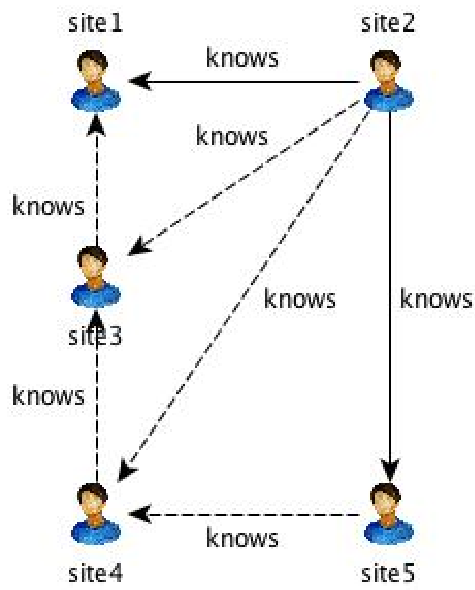


Figure 4: Social network extraction

can see some patches generated by *site*1, *site*3, *site*4 and *site*5 although it has only the capabilities from *site*1 and *site*5. Now *site*2 *knows* the existence of *site*3 and *site*4, moreover it can deduce the following: *site*3 *knows* *site*1, *site*4 *knows* *site*3 and *site*5 *knows* *site*4. So we can construct the local social graph of *site*2 as in figure 4(b). Where the solid-line arrows represent the direct relations between sites (by exchanging capabilities) while the dotted-line arrows represent the deduced relations which do not imply the ability to pull from these sites, since *site*2 has not the capabilities of these sites.

By applying this approach on all the sites, each site will have a local vision of his friend-of-friend network. We should take into consideration that the causal history keeps on growing and it will not delete previous entries. This means that the local vision at one site of the network keeps on growing as much as the site consumes from other sites. In the following section we present our validation and discuss the obtained results.

4. RESULTS

In order to compute the social network metrics with high confidence, we need a large dataset so we populated the DSMW ontology with the causal history of Mercurial repository for the Adium project [19], which follows a multi-synchronous interaction mode. The causal history that we analyzed is composed of 3128 patches (changesets in Mercurial terms) generated by 31 developers contributing to this project, over a period of 18 months. We were able to reveal 588 *knows* relations between the developers of this project. We calculated the key parameters of the social network graph, in order to compare it with a real social network graph parameters. In the following we will explain the obtained results.

A graph is considered small-world, if its average clustering coefficient is significantly higher than a random graph constructed on the same vertex set, and if the graph diameter is much smaller than the order of the graph [2]. The calculated parameters show that the extracted social network graph follow the small-world hypothesis.

Social network parameter	Calculated value
Network diameter	3
Clustering coefficient	0.634
Graph density	0.632

We also computed the centrality measures identified by [6] and illustrated the results using Gephi [3] the figures below show the obtained results. First we calculate the degree centrality which considers nodes with higher degrees as more central, highlighting the local popularity of an actor in its neighborhood. The way that we used to generate the social network graph created a directed graph so we have to calculate the inDegree centrality as shown in figure 5(a) (the number of incoming connections) and outDegree centrality as shown in figure 5(b) (the number of outgoing connections).

Then we calculated the betweenness centrality which focuses on the capacity of a node to be an intermediary between any two other nodes. The results are illustrated in figure 6(a). This figure shows clearly that we have one special actor in the network (most probably the project coordinator).

Finally we calculated the closeness centrality of the graph nodes which represents the node capacity to be reached by any other node in the network. Figure 6(b) shows the calculated closeness centrality.

(a) inDegree centrality

(b) outDegree centrality

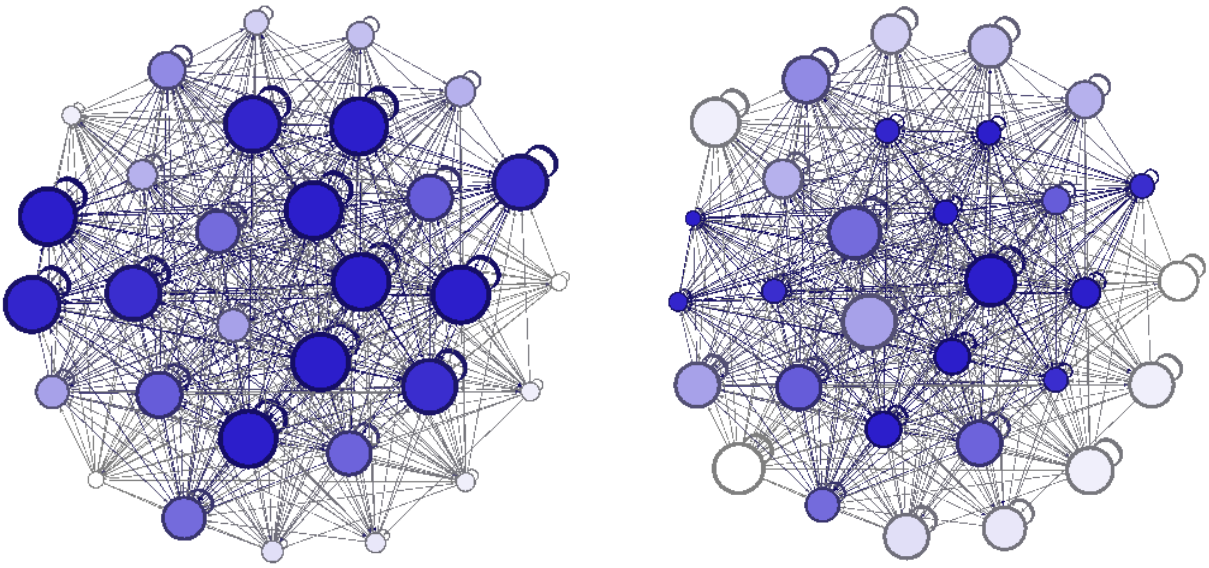


Figure 5: Degree centrality

(a) Betweenness centrality

(b) Closeness centrality

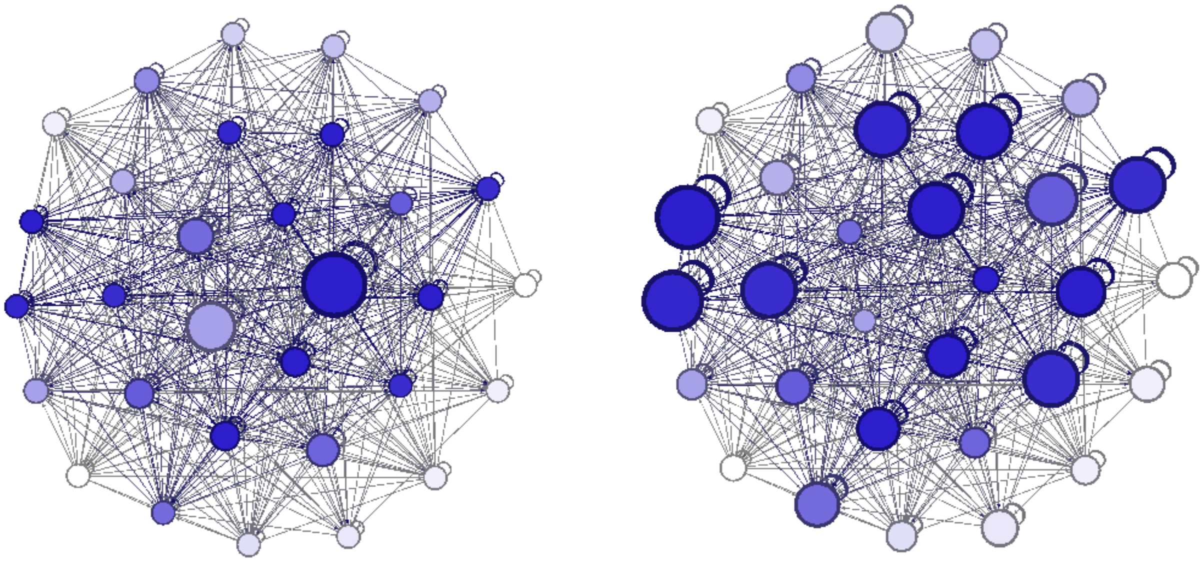


Figure 6: Betweenness and closeness centrality

From the results shown above we clearly see that we have extracted the local social network and based on the data we used, we see that the social network characteristics follow the small-world model, and we can see that we have some important actors in our network.

5. CONCLUSION

In this paper, we extended the collaborative services offered by multi-synchronous semantic wiki with new social services by exploiting the causal history using semantic queries. Therefore, each site participating in this collaborative network can have a local view of its social network without the need for a third-party service provider, and using its own resources.

We validated the approach using data from a software engineering application. And we found that the extracted network follows the small-world model. We will extend the validation furthermore, first over causal histories with more actors, then by applying the approach directly on DSMW sites as soon as we get enough data from such sites. Then we will integrate this social services directly inside DSMW. Finally we will develop the approach further by including the timestamps and the frequency of publishing/consuming among the different sites in order to quantize a proximity metric between sites.

6. REFERENCES

[1] L. Allen, G. Fernandez, K. Kane, D. Leblang, D. Minard, and J. Posner. ClearCase MultiSite: Supporting Geographically-Distributed Software Development. *Software Configuration Management: Scm-4 and Scm-5 Workshops: Selected Papers*, 1995.

[2] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of behavior of small-world networks, 2000.

[3] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An Open Source Software for Exploring and Manipulating Networks. In *International AAAI Conference on Weblogs and Social Media*. AAAI, 2009.

[4] P. Biddle, P. Engl, M. Peinado, and B. Willman. The darknet and the future of content distribution. In *In Proceedings of the 2002 ACM Workshop on Digital Rights Management*, 2002.

[5] M. Buffa, F. L. Gandon, G. Ereteo, P. Sander, and C. Faron. Sweetwiki: A semantic wiki. *Journal of Web Semantics*, 6(1):84–97, 2008.

[6] L. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[7] R. Gross, A. Acquisti, and H. J. Heinz, III. Information revelation and privacy in online social networks. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, New York, NY, USA, 2005. ACM.

[8] J. Hamano and L. Torvalds. Git - fast version control system. http://git.or.cz, 2005.

[9] A. Heinla, P. Kasesalu, and J. Tallinn. Skype. http://www.skype.com/, 2003.

[10] M. Krötzsch, D. Vrandecic, and M. Völkel. Semantic mediawiki. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 935–942, 2006.

[11] L. Lamport. Times, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7):558–565, July 1978.

[12] H. M. Levy. *Capability- and Object-Based System Concepts*, chapter 1. Digital Press, 1984.

[13] M. Mackall. Mercurial- a fast, lightweight source control management system. http://mercurial.selenic.com/, 2005.

[14] Microsoft. Microsoft office groove. http://office.microsoft.com/en-us/groove/, 2005.

[15] L. Miller and D. Brickley. The friend of a friend (foaf) project. http://www.foaf-project.org/, 2000.

[16] C. Rahhal, H. Skaf-Molli, P. Molli, and S. Weiss. Multi-synchronous collaborative semantic wikis. In G. Vossen, D. D. E. Long, and J. X. Yu, editors, *WISE*, volume 5802 of *Lecture Notes in Computer Science*, pages 115–129. Springer, 2009.

[17] M. Rogers and S. Bhatti. How to disappear completely: A survey of private peer-to-peer networks. In *SPACE 2007*, 2007.

[18] S. Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. *1st Workshop on Semantic Technologies in Collaborative Applications*, 2006.

[19] E. Schoenberg and Z. West. Adium : Instant messaging client. http://www.adium.im/, 2001.

[20] Skaf and et al. Dsmw : Distributed semantic media wiki. http://www.dsmw.org/, 2009.

[21] H. Skaf-Molli, C. Rahhal, and P. Molli. Peer-to-peer semantic wikis. In *20th International Conference on Database and Expert Systems Applications- DEXA 2009, Lecture Notes in Computer Science 5690, Springer*, volume 5690 of *Lecture Notes in Computer Science*, Linz, Austria, August 2009. Springer.

[22] S. Weiss, P. Urso, and P. Molli. Logoot-undo: Distributed collaborative editing system on p2p networks. *IEEE Transactions on Parallel and Distributed Systems*, 99(PrePrints), 2009.