Exposing Privacy Obligation Policies in Social Networking Sites

Paul Groth
Information Sciences Institute
University of Southern California
pgroth@isi.edu

Abstract

Increasingly, web-based applications are created through the composition of multiple functional components provided by different institutions. These so called "mash-ups" are an effective means to rapidly develop new applications. However, when these mash-ups are embedded within social networking sites that aggregate and expose personal data, such as Facebook, MySpace, or LinkedIn, serious privacy issues arise because personal data can be transmitted outside the applications hosting institution. In this paper, we describe an initial architecture and implementation to address these privacy concerns through the exposure of privacy obligation policies to the user using a workflow-based representation of mash-ups.

Introduction

Social networking sites provide an easy to use mechanism for people to keep in contact with their friends, colleagues, and family. Because of the viral growth of these sites (Facebook for instance currently has 59 millions users and is growing by 250,000 users a day (Facebook 2007)), a font of new personal information has been created. This information provides a valuable new resource for organisations to provide unique highly targeted products, services and commercial messages to users. The sharing of personal information is the core of the contract that social network sites make with users: by giving up some of their personal information to the site, users are provided with an easy means to build and maintain their relationships.

To encourage users to accept this contract and to diminish their legal liability, social network sites provide a form of *access control* to users, which allows users to specify who can see their various pieces of personal information. For example, a user could specify that only persons identified as their "friends" can see photos they have uploaded. As social networking sites have begun to exploit user's personal information, there have been increasing privacy concerns and some sites have been forced to withdraw planned functionality. For example, Facebook had to remove parts of its new purchase notification system, Beacon, after protests by its own users (Story and Stone 2007). Thus, ensuring that users can understand and control how their personal information is used is critical for the success of social networking sites.

To expand their business, social networking sites are increasingly allowing external providers to embed applications within their services. We term applications that are embedded within social networking sites, *social network applications*. Through application programming interfaces (API), Bebo, Facebook, Google-OpenSocial and others are allowing social network to offer additional functionality based on users' personal information. In terms of privacy, users must explicitly allow external providers to access their information. However, once a user permits access to their personal information, the application can use it as it pleases (subject to legal terms). The user has no mechanism to easily understand and specify *how* the application can use their data. In summary, within social networking sites their are no current mechanisms for exposing and specifying what Mont refers to as *privacy obligation policies* (Mont and Beato 2007).

The lack of these policies is of particular concern because of the *mash-up development approach* commonly used by providers of social network applications (Nickull, Hinchcliff, and Governor 2008). Mash-ups are web-based applications composed from multiple components (e.g. Web Services) provided by different institutions. Because components are provided by a variety of different institutions, personal data may be sent to an institution or processed and stored in a manner that the user disapproves of. This poses a risk to both social networking sites and external providers. If users find that their personal information is being improperly used or provided to unapproved parties, they could reduce or eliminate the information they provide to the service. Through the exposure of privacy obligation policies to users, we aim to address this privacy concern and encourage users to provide even more personal information enabling the development of highly tailored social network applications. Concretely, the contributions of this paper are as follows:

1. An architecture for the exposure of privacy obligation policies in social network centric mash-ups.
2. An initial implementation of the architecture integrated with the social networking site Facebook.

The rest of this paper is organised as follows. We begin with a discussion of privacy concerns with respect to social networking sites. We then present a social network application use case that illustrates privacy issues not currently

addressed by access control policies. We then discuss how these issues can be addressed through a flexible architecture for the exposure of obligations policies tailored to applications developed as mash-ups. Next, we detail our initial implementation of the architecture integrated with the social networking site, Facebook. Finally, we discuss how Semantic Web technologies might prove beneficial in this context and conclude.

Privacy Concerns in Social Networks

Privacy concerns in social networking sites have been recently received wide coverage in the news media (Stross 2007; Story and Stone 2007). These articles in particular have highlighted concerns over institutional (employers, health insurance companies, goverment) knowledge of people's private activities. However, although this media attention has increased users' awareness of the privacy implications inherent in the use of social networking sites, a significant number of users are unconcerned, unaware, or misunderstand how broadly their personal information is available online (Madden et al. 2007). Beyond institutions finding out personal information about their members, there are other significant privacy concerns as highlighted by Gross and Acquisti's study of 4000 Carnegie Mellon University students Facebook profiles[1] (Gross, Acquisti, and H. John Heinz 2005). This study identified a number of privacy risks including real-world stalking, online stalking, demographic re-identification, and face re-identification.

The Organization for Economic Cooperation and Development has identified eight principles for the protection of privacy that underlie the privacy laws of several countries including the US (OECD 1980). There are two principles in which social networking sites, in our opinion, do not currently provide enough support, these are *purpose specification* and *use limitation*. While these sites do provide access control mechanisms, the purpose and usage of personal information is not readily apparent to users and is usually specified in the privacy policy of the site. Such privacy policies are notoriously difficult for users to understand and are drafted specifically to protect companies from legal liability (Pollach 2007). This lack of knowledge is further amplified when a user uses multiple social networks applications each with their own terms of service and privacy policies, which may be implemented through the combination of services provided by various institutions. To illustrate this point, we present the following use case.

The Health News Use Case

A health insurance company aiming to improve the fitness off its customers, sponsors a social network application to provide personalised health related news. The developer of the application combines data from the user's profile, including their health interests, gender and location to filter health news and information from three different sources. As part of the application, the developer provides the health insurance company with highly targeted demographic information about the application's users. Figure 1 shows the Facebook page presented by our implementation of the application.

The news page generated by the application does not show the user how their data is processed and what institutions are involved in that processing. Fundamentally, *the user lacks the ability to understand and control the usage of their personal information once they have added the application to their profile.* Additionally, their is currently no generic means for applications to offer a mechanism that allows users to specify the obligations that the application should follow when making use of their personal information. To help address this deficiency, we now present an architecture for the exposure of obligation policies.

An Architecture for Policy Exposure

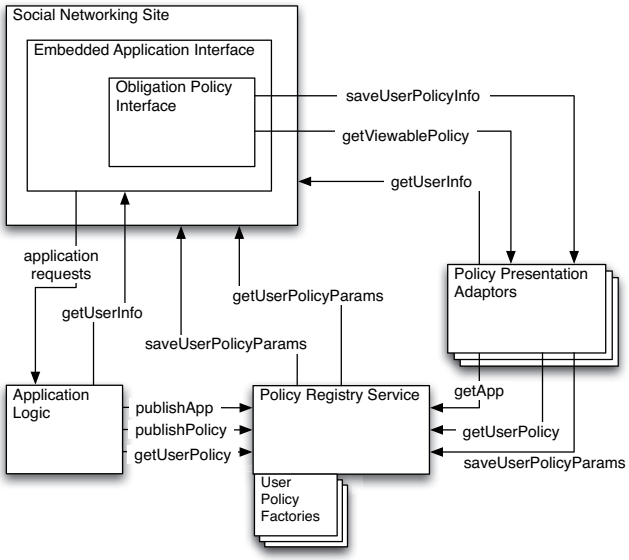

Figure 2: APES: Architecture for Policy Exposure in Social networks.

In designing our architecture for privacy obligation policies, we took a number of requirements into consideration. First, there are over 7000 different applications built on the Facebook platform (Facebook 2007). Thus, our architecture should cater for a variety of implementation techniques and be easy to integrate with existing applications. Second, there already exists a number of policy enforcement and management systems. The aim of our architecture is not to replace these existing systems but to augment them by allowing them to integrate with social networking applications. Third, the difficulty of understanding privacy policies has been noted as a deficiency in web sites, hence, the architecture should cater for intuitive user interfaces for displaying policies. Fourth, the architecture should not only allow for the exposure of obligation policies but other types of policies as well. Finally, given the quantity of users within social networking sites, the architecture should be designed with scalability in mind.

[1] A profile is the representation of the user, their details, and relationships within a social networking site.

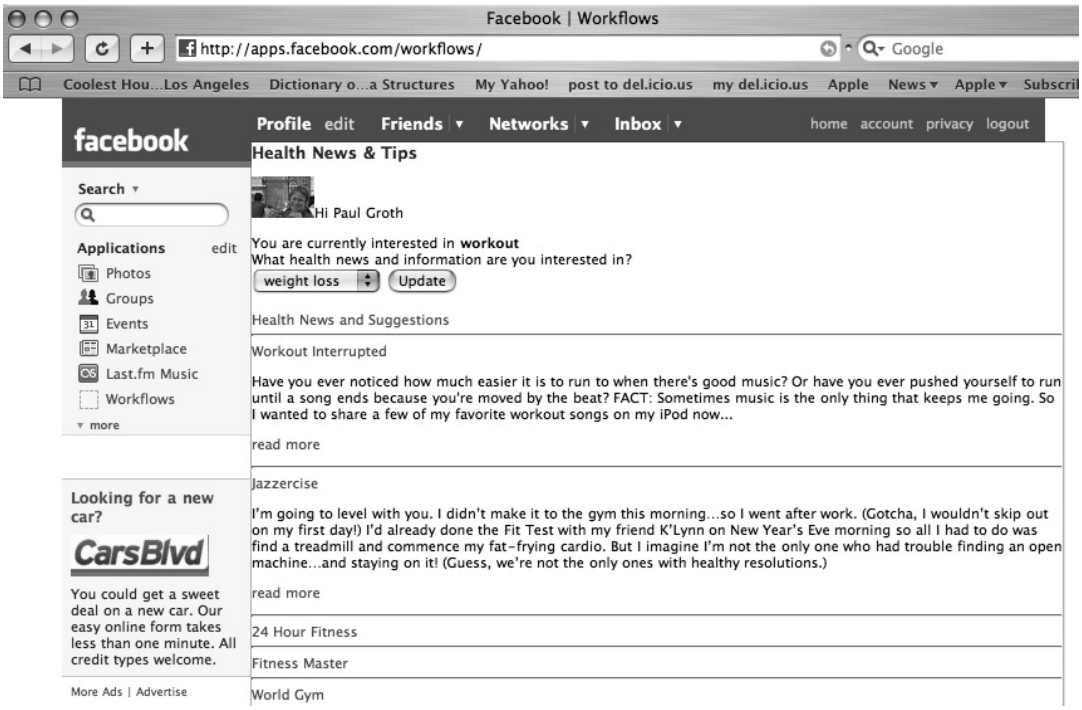

Figure 1: The Health News application.

Based on these requirements, we have designed a flexible service-oriented architecture shown in Figure 2. The basis of the architecture is the binding of an application representation and a corresponding policy for that application. The application representation is different from the application implementation. It is a model of the application at a level that end-users can understand. We have chosen to represent applications as workflows, which provide a high level description an of application's components and the data flow between them. Workflows were chosen as the representation for two reasons. One, are specific aim was to target social networking applications developed as mashups. Workflows naturally represent the composition of components inherent in mashups. Additionally, mashups are often built as workflows using editiors such as Yahoo's Pipes[2] and Microsoft's Popfly[3]. Second, prior work has shown that policies related to workflows can be used as a means for the representation and verification of privacy obligation policies (Cheung and Gil 2007). While the use of workflows is clearly better suited to the representation of service or component based, experience in prior work leads us to believe that applications built using different software engineering approaches can also be mapped to a workflow style representation (Groth, Miles, and Moreau 2008).

The selection of this particular application representation does not limit the underlying language used to express obligation policies. As we later discuss, the architecture is designed to support multiple policy languages and enforcement mechanisms through the use of adaptors. We now discuss each component of the Architecture for Policy Exposure in Social networks (APES), which is depicted in Figure 2.

Social Networking Site - The architecture assumes the social networking site has an API for embedding external applications as well as acquiring user information. We also assume that the site supports the storage of basic information generated by embedded applications on a per user basis. Both the Facebook and OpenSocial APIs support such persistence functionality.

Embedded Application Interface - The interface of the application that appears within the social network.

Application Logic -This architecture component represents the logic of the social network application. Typically, this is implemented as a web service. The application logic is responsible for publishing both the policies that it supports as well as a policy-compatible representation of its functionality as a workflow within the Policy Registry Service. The application logic is also responsible for the enforcement of policies, which can beachieved through integration with existing policy systems.

Policy Registry Service - This component is responsible for storing all application policies and application representations. Following from the work of Mont (Mont and Beato 2007), instead of storing a policy for every user, which could lead to large overheads especially for complicated policies, the architecture specifies that the Registry should generate user specific policies at runtime. This runtime generation is accomplished by storing user policy parameters using the

[2]http://pipes.yahoo.com

[3]www.popfly.ms

social networking site's persistence APIs[4]. The Registry caters for multiple different policy languages through a set of pluggable factories (User Policy Factories in the figure) that generate user policies based on the application policy and the user's policy parameters. Hence, when the application logic, responds to an application request it can obtain a specific user policy from the registry.

Policy Presentation Adaptors - To maintain our architectures flexibility, we introduce Policy Presentation Adaptors, which take a user policy and a a description of the application and generates a viewable policy for display. Thus, multiple policy language implementations can be used while still providing a common visual appearance to the user. Our aim is to provide an API in multiple programming languages to facilitate the creation of these adaptors.

Obligation Policy Interface - The last component of the APES architecture is the interface that renders viewable polices to the user. The interface should support the display of the application's functionality, the parties responsible for each of the pieces of functionality, along with the ability to change policies for each of the application's components.

The APES is designed to support multiple policy languages and implementations through the separation of application representation from policy description as well as the emphasis on a pluggable design both for user policy generation and the policy presentation. We now briefly discuss an initial implementation of this architecture integrated with Facebook.

Initial Implementation

Our initial implementation is written in PHP. We use a simple XML based policy that allows for the enabling and disabling of application components. The policy links to a high level description of the application as an XML workflow. The description contains the application component name, description, and the components inputs and outputs. Figure 4 shows the privacy settings page that our implementation generates for the Health News application from the combination of the application description, policy, and user's personal privacy preferences stored using the Facebook Data Store API. It is important to note that in this scenario the user must make a trade off between the functionality made available by the application and the distribution of their private information. For example, when disabling the Demographic Collector, the application uses defaults for the Local Search component, which will return results that are not pertinent to the user.

The developer of the application must also ensure that the various application components can be enabled and disabled without impacting the entirety of the application. The architecture caters for this by enabling applications to be described at a high-level, thus, the developer can ensure that the user is not disabling the application completely. For example, one can imagine that if the user disables all applications components, the application would still be able to return a status message.

[4]A persistence API is provided by both Facebook and OpenSocial.

The actual Health News application is a combination of a Facebook page rendered using PHP with the backend implemented using Yahoo Pipes (see Figure 3). The backend is called by the Facebook page which provides it with the user's name, health news interest, home location, date of birth, and gender. Using this information it takes data from RSS feeds provided by Mens Health and Womens Health magazines as well as medical news provided by Yahoo and filters them based on the gender and health interests of the user. This data is combined with a Local Search of businesses that cater to the particular health interest of the user in their geographic location. The combined data is published as an RSS feed that is then rendered as a Facebook page. Furthermore, the application supplies all the personal information to a web service, which in our scenario is operated by the health insurance company. Depending on the policy set, the Facebook page calls a different backend to implement the required policy.

APES and Semantics

The APES architecture relies upon two key data structures: the application representation and the policy. While the architecture is designed to cater for multiple implementations of these data structures, we believe that the adoption of Semantic Web technologies would be a beneficial implementation approach. In terms of policies, in a comparison of policy languages (Duma, Herzog, and Shahmehri 2007), those that were ontology based were able to cover a wide range of use cases including access control, usage control, and policy protection while still being declarative and flexible.

In addition, if a policy is described using Semantic Web standards such as RDF and OWL, the policy can more easily refer to entities in the application representation, especially if that representation also uses these standards. For example, if a policy restricted an application components of type *Search* from accessing a user's location and the *Local Search* component of the Health News application was a subclass of *Search* then the policy would apply. Thus, through the use of these standards, policies could be written that could apply to more than one application.

Beyond integrating more easily with policies, the use of Semantic Web technologies for application representations also helps with their display to the user. Many times workflows can be complicated by conversion routines, non core components or low-level details (e.g. the Union component in Figure 3). One of the functions of the Policy Presentation Adapter in the architecture is provide functionality for converting complex workflows to user specific representations. Through the use of ontologies, one can imagine being able to express that several components are part of a larger user understandable component. By making such relations explicit, the Policy Presentation Adapter would be easier to implement. One of our next steps is to investigate whether these hypothesized benefits translate into our concrete implementation.

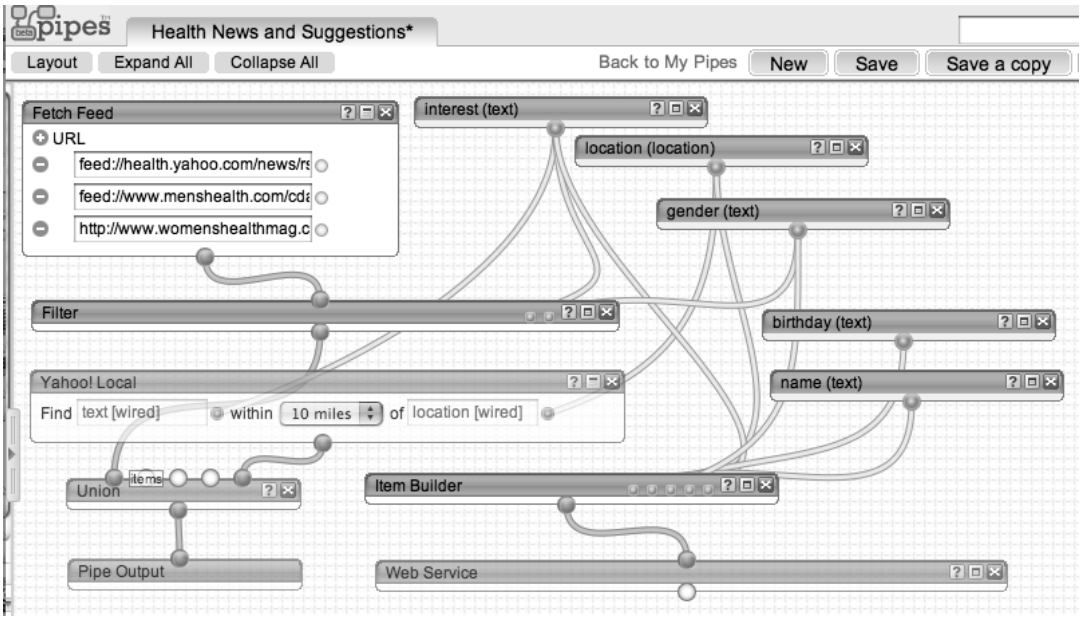

Figure 3: Yahoo Pipes workflow for the Health News Application.

Figure 4: Privacy settings page.

Conclusion

In this paper, we have presented an architecture and initial implementation for the exposure of obligation policies within social networking sites. In the future, we aim to further expand our implementation through the integration with more mature policy languages and systems. Furthermore, we aim to adopt a more detailed workflow representation from an already existing workflow system (Cheung and Gil 2007). Using our implementation as a basis, we are beginning to investigate the enforcement of policies in this domain through policy verification during and after application execution. We believe that policies for social network applications and mashups deserve attention by the community. There are numerous challenges in this domain where the community could apply its expertise and possibly impact the way that millions of people use these sites everyday.

References

Cheung, W. K., and Gil, Y. 2007. Towards Privacy Aware Data Analysis Workflows in e-Science. In *2007 Workshop on Semantic e-Science (SeS2007)*.

Duma, C.; Herzog, A.; and Shahmehri, N. 2007. Privacy in the semantic web: What policy languages have to offer. In *Eighth IEEE International Workshop on Policies for Distributed Systems and Networks, 2007 (POLICY '07)*, 109–118.

Facebook. 2007. Facebook Statistics Page. http://www.facebook.com/press/info.php?statistics. accessed Jan. 7, 2007.

Gross, R.; Acquisti, A.; and H. John Heinz, I. 2005. Information revelation and privacy in online social networks. In *WPES '05: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, 71–80. New York, NY, USA: ACM.

Groth, P.; Miles, S.; and Moreau, L. 2008. A Model of Process Documentation to Determine Provenance in Mashups. *Transactions on Internet Technology (TOIT)*.

Madden, M.; Fox, S.; Smith, A.; and Vitak, J. 2007. Digital Footprints: online identity management and search in the age of transparency. Technical report, Pew Internet & American Life Project.

Mont, M. C., and Beato, F. 2007. On Parametric Obligation Policies: Enabling Privacy-aware Information Lifecycle Management in Enterprises. In *Eigth IEEE Internation Workshop on Policies for Distrubted Systems and Networks (POLICY'07)*.

Nickull, D.; Hinchcliff, D.; and Governor, J. 2008. *Web 2.0 Patterns*. O'Reilly.

OECD. 1980. OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. Technical report, Organisation for Economic Co-Operation and Development.

Pollach, I. 2007. What's Wroing with Online Privacy Policies? *Communications of the ACM* 50(9):104–108.

Story, L., and Stone, B. 2007. Facebook Retreats on Online Tracking. New York Times.

Stross, R. 2007. How to Lose Your Job on Your Own Time. New York Times.