Privacy Stories: Confidence in Privacy Behaviors through End User Programming

Luke Church
University of Cambridge
Computer Laboratory
luke@church.name

Jonathan Anderson
University of Cambridge
Computer Laboratory
jonathan.anderson@cl.cam.ac.uk

Joseph Bonneau
University of Cambridge
Computer Laboratory
joseph.bonneau@cl.cam.ac.uk

Frank Stajano
University of Cambridge
Computer Laboratory
frank.stajano@cl.cam.ac.uk

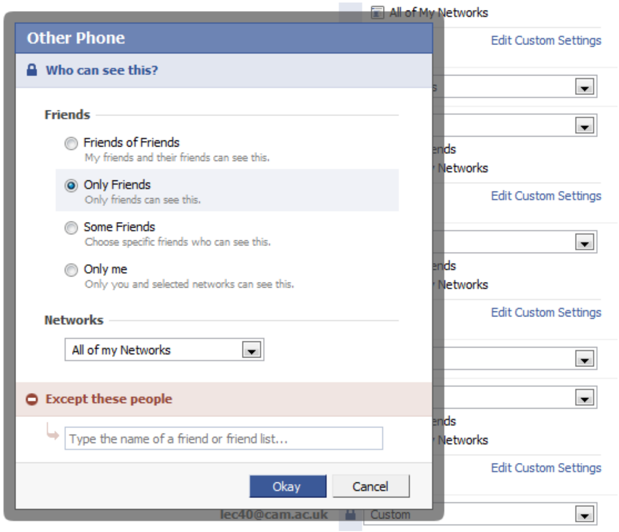
1. INTRODUCTION

In [2] we argued that, in the search to give users meaningful control over their information, we should consider End User Programming techniques as a possible replacement for either opaque, expert determined choices or the endless proliferation of options that arises from a simplistic application of direct manipulation principles.

We describe a *work in progress* to study the viability of this approach for improving the usability of social network privacy configuration. As suggested in [2] we make use of analytical usability techniques to discuss the usability challenges of the current Facebook interface and to inform the design of our proposed alternative. We then report on a very small (two-user) pilot study and look at challenges that we will address in future design iterations.

2. FACEBOOK'S PRIVACY UI

The current Facebook configuration UI, consists of a large number of configuration options (in excess of 60 at the time of writing, excluding configuration for photos), which are editable via screens such as that shown in Figure 1.



Other Phone

Who can see this?

Friends

○ Friends of Friends
My friends and their friends can see this.

● Only Friends
Only friends can see this.

○ Some Friends
Choose specific friends who can see this.

○ Only me
Only you and selected networks can see this.

Networks

All of my Networks

Except these people

Type the name of a friend or friend list...

Okay Cancel

Figure 1 – Part of Facebook's Privacy Configuration UI

Applying a Cognitive Dimensions [4] analysis predicts a number of difficulties here: the user interface is remarkably *Diffuse*$_{CD}$, having such a large number of options risks the user missing options. It also makes the interface very *Viscous*$_{CD}$ – changes become very time consuming, increasing the total attention cost [1], to a level that the user may be unwilling to spend in order to manage the *Secondary Goal* [8] of managing privacy. These problems of diffuseness and viscosity affected both of our users in the pilot study.

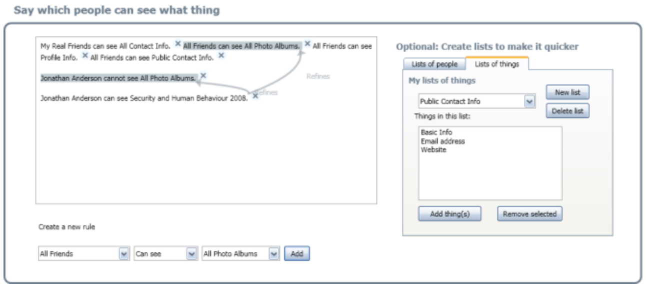
The UI also has potentially poor *Role Expressiveness*$_{CD}$. Many of the lists, such as 'Networks' were not specifically created by the user, and do not necessarily correspond to concepts that the user would be familiar with. Again, this caused problems for both users in our pilot study.

Furthermore, the UI has poor support for *Abstraction*$_{CD}$, it is unclear how to define new abstractions, which abstractions are public and which are private, *Hidden Dependencies*$_{CD}$, caused by the interaction of the abstractions, high *Premature Commitment*$_{CD}$, of forcing users to make abstraction decisions well ahead of the time when they are used.

Whilst this might appear to be highly critical of Facebook's UI in particular, this is not really the case. Access control systems are difficult to design in a way in which they can be successfully used by experts, let alone end users. In the light of this, the critique is more a starting position for research rather than a critique of Facebook per-se.

3. PRIVACY STORIES

We propose an alternative interaction style to address these problems. The user constructs a textual, programmatic representation of their privacy wishes. This allows for a much less diffuse interface, allowing all of the privacy and access management systems to be displayed on a single page. We provide explicit, optional, support for creating new abstractions, through lists of users and 'things', shown in Figure 2.



Say which people can see what thing

My Real Friends can see All Contact Info. ✕ All Friends can see All Photo Albums. ✕ All Friends can see Profile Info. ✕ All Friends can see Public Contact Info. ✕

Jonathan Anderson cannot see All Photo Albums. ✕

Jonathan Anderson can see Security and Human Behaviour 2008. ✕

Create a new rule

All Friends | Can see | All Photo Albums | Add

Optional: Create lists to make it quicker

Lists of people | Lists of things

My lists of things

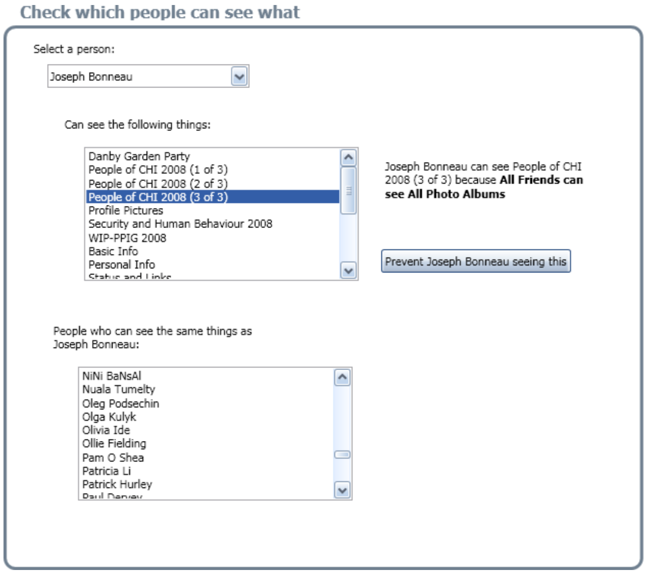
Public Contact Info

Things in this list:

Basic Info
Email address
Website

Figure 2 - Rules and Lists UI element

In an attempt to increase user confidence as to the behavior of their rules, we draw on work in 'End User Engineering' [3], providing testing and debugging functionality where users can validate who can see which elements of their profile, look at the equivalence class of which other users have the same access. Further, in an approach inspired by Ko's WhyLine [6], textual descriptions of why access behavior occurs are automatically provided, as shown in Figure 3.



Check which people can see what

Select a person:

Joseph Bonneau

Can see the following things:

- Danby Garden Party
- People of CHI 2008 (1 of 3)
- People of CHI 2008 (2 of 3)
- People of CHI 2008 (3 of 3)
- Profile Pictures
- Security and Human Behaviour 2008
- WIP-PPIG 2008
- Basic Info
- Personal Info
- Status and Links

Joseph Bonneau can see People of CHI 2008 (3 of 3) because **All Friends can see All Photo Albums**

People who can see the same things as Joseph Bonneau:

- NiNi BaNsAl
- Nuala Tumelty
- Oleg Podsechin
- Olga Kulyk
- Olivia Ide
- Ollie Fielding
- Pam O Shea
- Patricia Li
- Patrick Hurley

Figure 3 - Testing and debugging UI element

In order to guide the refinement of the design, we performed a small, two-user pilot study.

4. PILOT STUDY

Our study consisted of two users, a 32 year old male (User A), and a 27 year old female (User B). Both had some familiarity with office software, email and Facebook, but little programming experience. We interviewed them prior to the study, discussing their use of Facebook, and their sensitivity to the potential privacy issues of social networks. Both expressed only the mildest of concern over the control of the information. User A's articulated need for privacy was more protecting his information, especially his email address, from Facebook as a corporate entity rather than from other users.

They were then asked to attempt to express their privacy wishes in each interface, in both cases using their real data, User A starting with Facebook, User B starting with our 'Privacy Stories' interface. A think aloud protocol was used, followed by a discussion to attempt to determine their understanding of, and confidence in, the behavior provided by the two interfaces.

Both users reported higher confidence in their understanding of the behavior of the Privacy Stories interface. In User A's case, this seems to have come from the decrease in diffuseness; he could express his fairly simple requirements more tersely. In User B's case, this seemed to arise from the clearly visible ability to deal with concrete people. User B described the increased flexibility offered by the interface as contributing to her increased experience of control.

Despite this, the study could hardly be regarded as an endorsement of the current design. User A was comfortable with the default policy we provided, finding little need to refine it. User B's reasoning about the operations of lists was substantially incorrect, leading her to make a number of rules which didn't match her stated intention. Regrettably, she found the list manager more confusing than helpful. Both users found some of the non-standard aspects of the interaction paradigm, such as the read-only equivalence view, confusing. However User A, once he understood the behavior of the tool, appeared to find it helpful and it supported his confidence in the system's behavior.

Whilst there were a number of problems, there were also a number of indications that support our general approach of using a programming-like style of interaction that allows users to express their specific wishes. For instance in our study we two users with very different privacy requirements. In a behavior reminiscent of House's descriptions of social obligations on Flickr [5**],** User B wanted to use the privacy management interface to manage how much information she was publishing about herself, *to decrease her impact on other peoples' experiences*, not to regulate how much other people could find out about her. User A wanted to regulate in detail only a very specific piece of information, his email address, but was less concerned with other details. Such disparate requirements would be difficult to capture in an 'expert defaults' system. Further, the act of constructing the policies seemed to have a substantial effect increasing the confidence of both users in the behavior of the system, and finally, the attention cost of using our system appeared to be at least comparable to using Facebook's, but scaled better to more complex behaviors.

This is not good enough; both subjects made it clear that the attention costs of both interfaces were too high for them to have bothered. With the Facebook style of interface this is a hard problem, if anything the number of options is increasing over time, however we hope that with the introduction of a more gentle slope of abstraction [7], the Privacy Stories interface will achieve a lower attentional cost.

We suggest then, that the Privacy Stories interface shows a need for further refinement, especially around the handling of abstractions, but appears to hold potential for improving users' control and confidence in their privacy choices.

5. REFERENCES

[1] Blackwell, A.F. 2002. First steps in programming: A rationale for Attention Investment models. Proceedings of IEEE Symposia on Human-Centric Computing Languages and Environments, pp 2-10

[2] Church, L. and Whitten, A. 2009. Generative Usability: Security and User Centered Design beyond the Appliance. *Submitted*: New Security Paradigms Workshop, Oxford 2009

[3] EUSES Consortium, eusesconsortium.org, visited on 26/05/2009

[4] Green, T. and Petre, M. 1996. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. Journal of Visual Languages and Computing 7, 2, pp 131-174

[5] House, N. A. 2007. Flickr and Public Image-Sharing: Distant Closeness and Photo Exhibition in CHI 2007, pp 2712-2722

[6] Ko, A.J. and Myers, B. A. Designing the Whyline, A Debugging Interface for Asking Why and Why Not questions about Runtime failures in CHI. 2004, pp 151-158

[7] Pane, J. and Myers, B. 2006. Natural programming languages and environments. In End User Development pp 31-50

[8] Whitten, A. Making Security Usable. 2004. PhD Thesis, Carnegie Mellon University