Carnegie Mellon University
Research Showcase @ CMU

Human-Computer Interaction Institute | School of Computer Science

2007

Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application

Norman Sadeh
Carnegie Mellon University

Jason Hong
Carnegie Mellon University

Lorrie Cranor
Carnegie Mellon University

Ian Fette
Carnegie Mellon University

Madhu Prabaker
Carnegie Mellon University

See next page for additional authors

Follow this and additional works at: http://repository.cmu.edu/hcii

Recommended Citation
.

This Article is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Human-Computer Interaction Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Authors
Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Madhu Prabaker, and Jinghai Rao

This article is available at Research Showcase @ CMU: http://repository.cmu.edu/hcii/80

Norman Sadeh · Jason Hong · Lorrie Cranor · Ian Fette · Patrick Kelley · Madhu Prabaker · Jinghai Rao

Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application

Received: date / Accepted: date

Abstract A number of mobile applications have emerged that allow users to locate one another. However, people have expressed concerns about the privacy implications associated with this class of software, suggesting that broad adoption may only happen to the extent that these concerns are adequately addressed. In this article, we report on our work on PEOPLEFINDER, an application that enables cell phone and laptop users to selectively share their locations with others (e.g. friends, family, and colleagues). The objective of our work has been to better understand people's attitudes and behaviors towards privacy as they interact with such an application, and to explore technologies that empower users to more effectively and efficiently specify their privacy preferences (or "policies"). These technologies include user interfaces for specifying rules and auditing disclosures, as well as machine learning techniques to refine user policies based on their feedback. We present evaluations of these technologies in the context of one laboratory study and three field studies.

Norman Sadeh
ISR - School of Computer Science - Carnegie Mellon University
5000 Forbes Avenue - Pittsburgh PA 15213-3891
sadeh@cs.cmu.edu

1. Introduction

Over the past few years, a number of mobile applications have emerged that allow users to locate one another. Some of these applications are driven by a desire from enterprises to increase the productivity of their employees. Others are geared towards supporting social networking scenarios, such as meeting up with friends, or safety-oriented scenarios, such as making sure that a loved one returned home safely. The growing number of cell phones sold with location tracking technologies such as GPS or Assisted GPS ("A-GPS") along with the emergence of WiFi-based location tracking solutions could lead to mainstream adoption of some of these applications.

In this article, we report on work conducted at Carnegie Mellon University in the context of PEOPLEFINDER, an application that enables cell phone and laptop users to selectively share their locations with others, such as friends, family, and colleagues (see Figure 1). This article extends a previous workshop paper in which we introduced PEOPLEFINDER [6], and provides a more thorough and detailed report of our user studies.

Our objective has been to better understand people's attitudes and behaviors towards privacy as they interact with such an application, and to explore technologies that empower users to more effectively and efficiently specify their privacy preferences (or "policies").

The work presented in this article confirms that people are generally apprehensive about the privacy implications associated with location tracking. It also shows that privacy preferences tend to be complex and depend on a variety of contextual attributes (e.g. relationship with requester, time of the day, where they are located). Through a series of user studies, we have found that most users are not good at articulating these preferences. The accuracy of the policies they define increases only marginally over time unless they are given tools that help them better understand how their policies behave in practice.

Our studies included a combination of controlled lab experiments with 19 users and field studies involving a total of over 60 participants. Our results suggest that functionality that increases user awareness can contribute to the definition of more accurate policies. As our users grew more comfortable with PEOPLEFINDER and the way in which it was used by their acquaintances, they started refining their preferences and relaxing some of their policies to allow for requests that would have been denied under their initial policies. This suggests that functionality that empowers users to more effectively control their policies can contribute to the adoption of context-aware applications like PEOPLEFINDER.

This article also compares results obtained in the context of controlled lab studies with results from longitudinal studies spanning up to several weeks. While both types of studies show that users have a hard time defining policies, our results suggest that users tend to be significantly more careful when defining policies that will be used to make decisions in actual situations (rather than under simulated conditions). To the best of our knowledge, the results from our field studies are the first of this type to analyze the behavior of users, the accuracy of their policies and how their policies evolve in the context of a fully deployed application with actual users.

Finally, we also investigate whether machine learning techniques can be effective in helping to increase the accuracy of user policies. Our early results suggest that these techniques are promising.

The remainder of this article is organized as follows. Section 2 gives a brief overview of related work. Section 3 provides an overview of our PEOPLEFINDER application. Section 4 discusses the privacy policy authoring functionality we have developed as well as several enhancements we are currently working on. An overview of PEOPLEFINDER's auditing functionality is provided in Section 5. Section 6 provides a summary of lab experiments we conducted in Summer 2006. Results and observations from a series of three pilot studies in which participants used the system as part of their daily lives in Spring 2007 are presented in Section 7. Section 8 has a discussion of results, and Section 9 presents some concluding remarks.

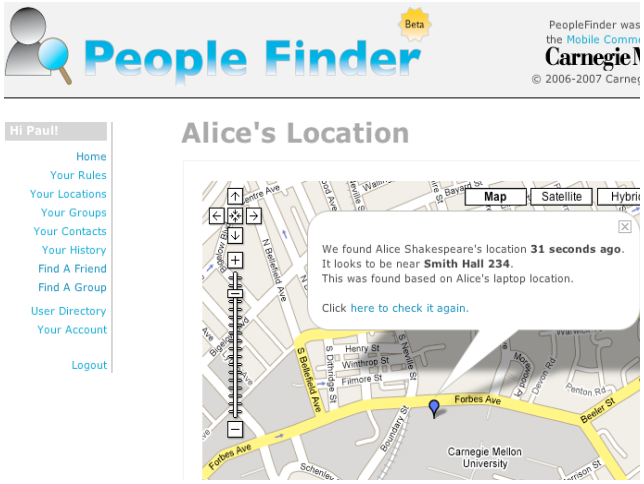

Figure 1. PEOPLEFINDER is an application that lets users share their locations with others subject to privacy policies they can refine over time.

2. Related Work

From a high-level perspective, past work on location privacy can be grouped into three categories: computational, architectural, and user interface. Our work is most related to the user interface category.

Computational approaches to location privacy propose algorithms for ensuring that disclosed data meets specified levels of privacy protection. Much of the work in this category strives to protect the anonymity of a set of users rather than a specific individual. More specifically, they typically aim for *k*-anonymity, in which disclosed data is anonymized such that there are at least *k* people sharing any combination of disclosed attributes. Thus, these algorithms strive to provide a guaranteed level of protection. Examples of this kind of work include Gruteser and Grunwald's work on spatial and temporal cloaking [8], and Beresford and Stajano's work on mix zones [3]. Other algorithmic approaches look at how to protect attackers from inferring more information about an individual, given a trace of that person's location information. For example, Krumm presented several techniques for determining the home address of an individual, given location data from volunteer users

[17]. Krumm provides a comprehensive overview of the state of the art in computational privacy [18].

Architectural approaches to location privacy are system architectures meant to limit what location information is collected and/or how that information can be queried. Two examples of research systems that focus on the former are Cricket [23] and Place Lab [19] which rely on "beacons" in the infrastructure to locally compute a user's current location. These systems are more privacy protective than systems in which users broadcast information that allows the system to compute their current location, as in Active Badges [28]. Hitchhiking is another approach which looks at how busy places are rather than looking up the location of any specific individual [27]. Hightower and Boriello have published a survey of techniques for determining one's location [10].

Other systems focus more on restricting how location information is processed and queried. For example, Hong and Landay presented a system that computed, processed, and accessed data locally as much as possible, minimizing access to any network resources and thus maintaining some notion of privacy [13]. Canny and Duan [4] and Rastogi et al. [24] have presented work that limits what information people can see to situations where they were physically present. Canny and Duan have taken a cryptographic approach, whereas Rastogi et al. have taken an Application Programming Interface approach.

There has been a fair amount of user interface work looking at what information people are willing to share under what conditions, in the form of diary studies [2], interviews [9, 12, 15], surveys [20], and experience sampling techniques [5, 16]. Surveys by Lederer et al. suggest that *who* is requesting the information is the primary factor in choosing whether to disclose information or not [20]. Consolvo et al. [5] saw similar results using experience sampling, finding that the most important factors regarding disclosures were *who* was requesting one's location, *why* that person was making the request, and *what level of detail* would be most useful to the requestor.

Other work has looked at the design and evaluation of working systems. This includes Reno, a mobile social system for sharing location information; MySpace[2], a system for managing privacy policies governing what others could see about one's location, availability, calendar information and instant messaging [22]; and IMBuddy, a sister project of PEOPLEFINDER that looked at sharing information about one's location, interruptibility, and active window in the context of an enhanced instant messenger client [14].

PEOPLEFINDER builds on this past work by looking more deeply at what types of privacy policies people have, how good they are at specifying these policies and how they can be supported in that process. This work has involved a combination of both lab studies and actual field deployments spanning up to several weeks. Our work is not limited to designing policy authoring and auditing interfaces. Instead, we also present results suggesting that machine learning techniques can help refine the accuracy of people's privacy policies.

3. Overview of PEOPLEFINDER

In PEOPLEFINDER, users rely on Policy Enforcing Agents (PEA) to handle queries about their locations (see Figure 2 and 3). The user's PEA operates according to a policy (or set of rules) specified by the user, with each rule granting access to the user's location under a particular set of conditions (e.g. query coming from a particular group of users on one of several possible days and within one of several possible time windows).

Users can invite other people (e.g. friends, family members, or colleagues) to check their location with PEOPLEFINDER, using either a mobile phone client or the PEOPLEFINDER web site. Users can specify rules under which other people can access their location and define groups of people to which particular rules apply.

PEOPLEFINDER is available for Windows Mobile cell phones and for both PC and Apple laptops. The cell phone version relies on GPS technology to pinpoint the user's location. When no GPS reading is available (e.g. the user is indoors), the application falls back on a GSM triangulation solution developed by Intel Research Seattle [26]. While the GSM approach is not as accurate as GPS, it provides an estimate of the user's location, often within a few hundred yards, under a wider set of conditions.

The laptop version uses a WiFi positioning solution developed by Skyhook Wireless [29]. In urban areas, this solution tends to have an accuracy of about 30 yards. It is complemented by an ad-hoc WiFi-based solution developed specifically for Carnegie Mellon's campus, which lets us estimate in what room a person is located when on campus. This latter solution, which uses a database of access points on campus, often provides readings that are even more accurate than the more general Skyhook Wireless solution.

[2] Not to be confused with the popular social networking site with the same name.

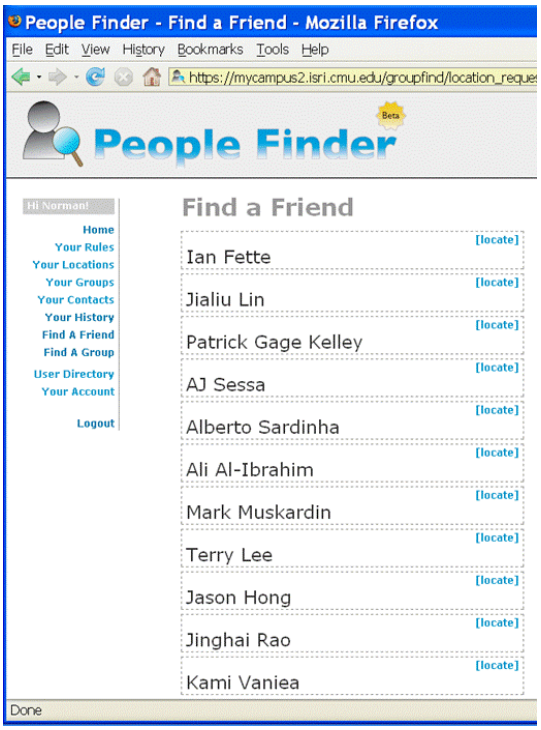

Figure 2. Browser-based interface for finding the location of a person. Equivalent Java and C# applications are also available for laptops and several cell phones.

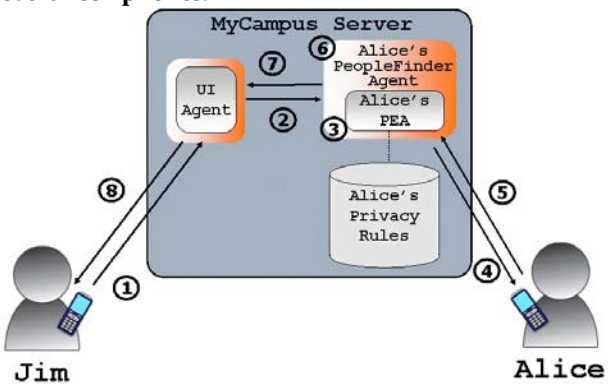

Figure 3. Processing Jim's request for Alice's location.

To understand how PEOPLEFINDER works, we need to distinguish between *target users*, namely users who are willing to share their locations with others, and *requesting users*, namely users who can submit queries about the location of one or more target users. A user can be both a target user and a requesting user, but does not have to be. Target users who rely on their laptops to track their location need to download an application on their laptops. This same application can be used for quickly finding a person as well as getting feedback about requests made about your location (discussed later below, see Figure 6). J2ME and C# versions of the application have also been developed for target users who rely on their cell phones to track their location, though these versions only work on a limited number of smartphone models. The smartphone version also lets users query for other people's locations.

Figure 3 outlines the main steps involved in processing a query from a requesting user (Jim) for the location of a target user (Alice). The request submitted by Jim is forwarded by his User Interface Agent (e.g. Web browser or cell-phone application) to Alice's PEOPLEFINDER Agent. The agent invokes Alice's Policy Enforcing Agent (PEA) to check whether the query is consistent with the privacy rules specified in her policy. If it is, a notification is forwarded to Alice's location tracking device, a cell phone in this example. Also, Alice's phone periodically updates her PEOPLEFINDER agent with her current location, regardless of whether anyone is requesting it. This design decision makes it faster for requesting users to see location information, as well as letting us provide a "last seen" functionality. Once returned, the location may need to be further processed by Alice's PEOPLEFINDER Agent (e.g. to combine multiple readings of Alice's location such as a GPS reading from a few minutes ago and a more recent reading based on GSM triangulation) before being forwarded to Jim. Finally, the results of the request are displayed on Jim's client, as shown in Figure 1.

When a request for a target user's location cannot be satisfied, PEOPLEFINDER returns an ambiguous message that does not allow the requester to determine whether the request was denied by the target user's policy or whether the target user's device (laptop or cell phone) could not be found. This provides a basic level of plausible deniability, in that a target user could claim to have forgotten to run the application, had her laptop off, or was simply out of range even if she actually had a rule in place blocking disclosure in this particular instance.

In general, processing may be somewhat more complex and some privacy rules may in fact require checking Alice's location to determine whether or not to disclose her location. For instance, Alice may have specified that her colleagues can only access her location during weekdays and while she is on campus. Query processing could also involve the use of obfuscation rules that manipulate the accuracy of the response returned to a user [3, 25].

PEOPLEFINDER is built on top of the MyCampus infrastructure, a semantic web environment in which policies are expressed using a rule extension of the OWL language [25]. The resulting language is capable of modeling a wide range of policies. Access to a

user's location can be restricted according to conditions that refer to any number of concepts or instances of concepts defined in an open collection of ontologies (e.g. ontologies of locations, social relationships, and calendar activities). This includes capturing a variety of context-sensitive restrictions such as disclosing your location only when you are in a particular place, or enforcing obfuscation policies that allow users to specify how they want the application to manipulate the accuracy of their location before disclosing it (e.g. city-level versus street address).

Presently, PEOPLEFINDER only uses a small fraction of the policies that can be expressed in this framework. In fact, one of the questions our project is attempting to address has to do with how much expressiveness is actually required for users to feel comfortable using the application and to what extent adding more expressiveness enables users to more accurately specify their policies – in contrast to creating more confusion.

Finally, as noted earlier, we opted to store location information centrally on our servers, rather than taking a decentralized approach as advocated by past work [13, 19, 26]. Again, this lets us provide a "last seen" functionality, but also made it much easier to quickly modify and update the system, an important consideration for rapid prototyping and for research. This centralized approach enabled us to develop thin clients on phones, with most of the functionality existing on our servers rather than on individual mobile devices. This decision made it easier for us to support a larger number of clients, an important consideration given the wide diversity of mobile phones in use today. However, in a centralized approach, the central server is a potential privacy vulnerability.

4. Privacy Policy Authoring

Users can define rules in which they grant access to their locations to individuals or groups of users. Each rule includes one or more restrictions such as the days of the week or times of day during which location queries from particular individuals or groups of users will be granted, as shown in Figure 4. Users can define user groups and place individuals in multiple groups.

Extensions of the rule interface also allow users to specify locations as collections of rectangles on a map (e.g. all buildings in the School of Computer Science) and specify rules that include location-based restrictions (e.g. only disclose my location when I am in a School of Computer Science building), as shown in Figure 5.

To avoid conflicts in rules, we currently allow only rules that grant access to one's location rather than combinations of rules with some granting access and others denying access. For example, a person can specify "Mary can see my location between 9AM and 5PM", but cannot specify, for example, "Colleagues cannot see my location on weekends".

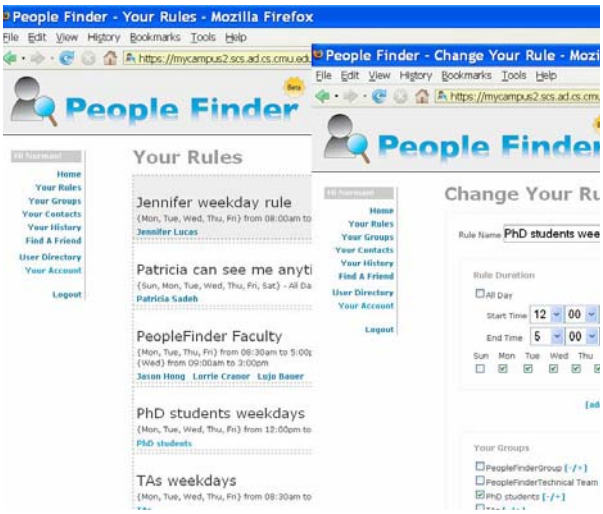

Figure 4. User interface for defining simple privacy rules.

Figure 5. Users can also define locations as combinations of rectangular areas for use in location-sensitive privacy rules.

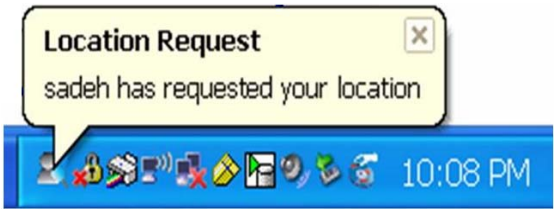

Figure 6. Bubbles notifying users of incoming queries help maintain awareness while being minimally disruptive.

5. Auditing Functionality

Results reported in Sections 6 and 7 show that users often have difficulty anticipating the conditions under which their peers will use the application to access their location. To be effective, user interfaces have to be designed to increase user understanding over time of how the application is being used. We have found that simple bubbles that discreetly pop up (e.g. at the bottom of a laptop screen) to notify users that their location is being requested can go a long way in helping users feel more comfortable with the application (see Figure 6). This feature is also intended to deter requesters from potentially abusing the system as they know their requests will be seen by target users – a similar feature has been used in IMBuddy, a sister project [14].

An even more important element is the design of auditing functionality that enables users to review requests that have been submitted, see how they were processed by the rules they currently have in place, and possibly request a more detailed explanation to identify rules they may want to modify.

In PEOPLEFINDER, users have a number of options to audit previously submitted requests. This includes reviewing requests that were denied or requests that have not yet been audited, as shown in Figure 7. They can incrementally access additional details about a particular request, such as where they were when their location was requested or the way in which their location was estimated (e.g. GPS versus GSM), as shown in Figure 8.

The interface also supports *explanation functionality*. As Figure 8 illustrates, the system identifies what rules led to a particular disclosure/non-disclosure decision. By letting users indicate whether they are satisfied with the decision made based on their current policy, the system can try to help users refine their policies. Sections 6 and 7 present results obtained by running different learning algorithms on the feedback obtained from users to help refine their policies. The same type of feedback could also be used to initiate dialogues and offer suggestions on how they could improve the accuracy of their rules.

6. Initial Lab Experiments

Our current version of PEOPLEFINDER reflects several design iterations with users. Initial work was conducted using a mockup application designed to present users

Figure 7. Auditing functionality helps users understand how their policies work and enables them to more effectively refine their policies.

Figure 8. Explanation can help users better understand their policies. User feedback can also be used to make suggestions or learn the user's preferences.

with scenarios that captured elements of their daily routines and interactions with members of their social networks. In this section, we briefly summarize findings from this initial work, which revolved around lab experiments involving 19 participants. In Section 7, we present more recent results from 3 pilot studies conducted with users of a deployed version of PEOPLEFINDER. This second set of experiments involved a total of over 60 participants. We discuss how results from the latter studies reinforce most of our initial findings and also point to a few differences between these two sets of experiments.

In our laboratory experiments, users were asked to provide information about their daily routines and social networks (e.g. names of key family members, boyfriend/girlfriend/spouse, colleagues/classmates, and friends). Each participant was asked to specify rules indicating the conditions under which she would be willing to share her location information with others (e.g. "My colleagues can only see my location on weekdays and only between 8am and 6pm"). The experiments involved presenting each participant with a total of 30 individualized scenarios (45 scenarios for each of the last 4 participants). Each individualized scenario included asking the participant whether she felt comfortable disclosing her location, showing her what her current policies would do, and offering her a chance to refine her policies.

On average, subjects required a little over 5 minutes to specify their initial rules and nearly 8 minutes if one includes the time spent refining their rules as they were confronted with new situations. Several users ended up with 8 or more rules by the end of the experiments. Despite the time and effort spent specifying and refining their policies, participants were generally unable to achieve high levels of accuracy. Based on feedback provided as they were presented with individualized scenarios, subjects indicated they were only satisfied with 59% of the decisions made by their initial rules, as shown in Figure 9. As they refined their rules over time, that percentage only went up to 65%. Even when using the rules that users ended up with at the end of the experiments and re-running these rules on all 30 (or 45) scenarios, decisions were only correct 70% of the time.

During the course of the experiments, most users refined their existing rules and added new ones, as shown in Figures 10a and 10b. In other words, the relatively small increase in policy accuracy (from 59% to 70%) cannot be attributed to a lack of effort from users in trying to refine their policies. Nor can it be attributed to a poorly designed interface. As can be seen in Figure 11, most users thought that the interface

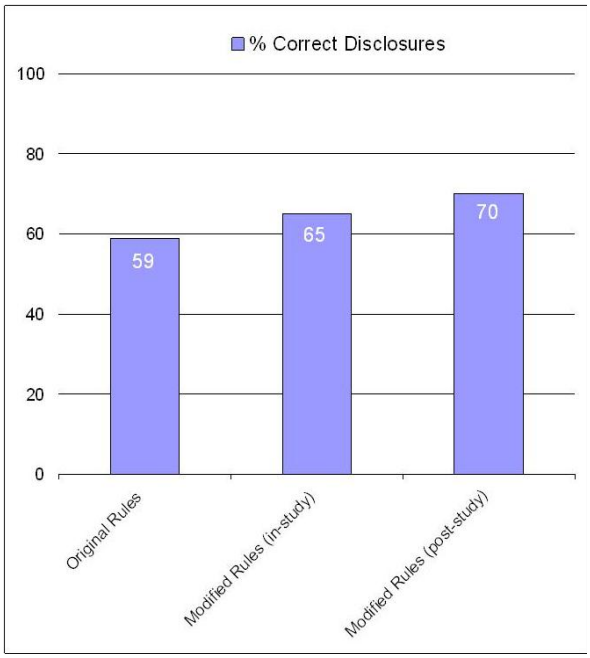

Figure 9. Controlled lab experiments: Users are not very good at articulating their privacy policies – accuracy of initial rules versus rules modified after being presented with 30 customized usage scenarios.

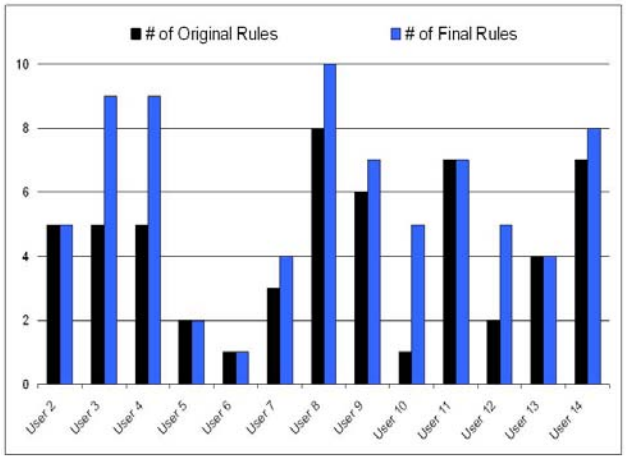

Figure 10a. Controlled lab experiments: initial number of rules versus final number of rules. User 1 was used for a pilot study and thus is not included in these results.

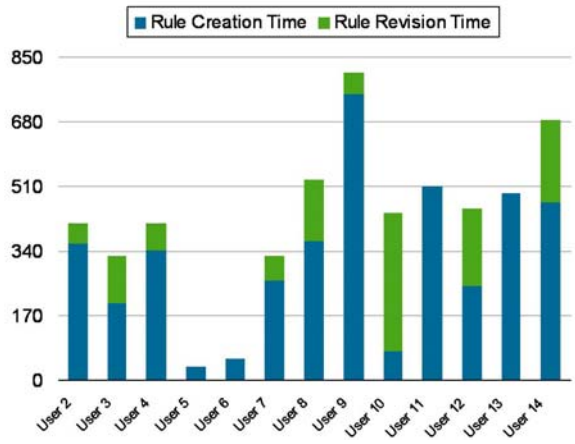

Figure 10b. Controlled lab experiments: time (in seconds) spent creating and modifying rules – the latter includes both changes to initial rules and addition of new rules.

for modifying their rules was easy to use.

In fact, there is relatively little correlation between policy accuracy and the number of rules specified by participants (see Figure 12). Similarly, there is little correlation between policy accuracy and the time spent by participants defining and refining their rules (see Figure 13). Instead, it seems that users reach a plateau and are often unable to articulate highly accurate policies.

While users seem to have a hard time accurately describing their privacy policies, their feedback tends to be fairly consistent and can be used as a basis for learning more accurate policies. Results displayed in Figure 14 compares the accuracy of policies defined by each of the 19 participants, examining the correctness of the participant's original rules, modified rules applied progressively (i.e., applied to all scenarios after the modification), and final rules as applied to all of the scenarios in a post hoc manner. Figure 14 also examines the effectiveness of applying case-based reasoning (CBR) using a *k*-nearest neighbor heuristic [1]. In this approach, each new situation is compared with prior cases available for a given user. The *k* closest cases cast a vote on whether to disclose the user's location or not (computed individually for each user). CBR systematically improved the accuracy of the policies to 82% (versus 70% when re-applying the user's final policies to each of the scenarios).

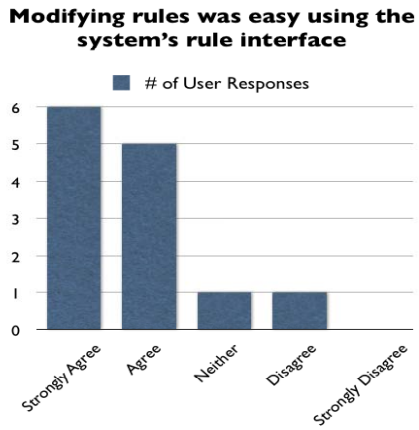

Figure 11. Controlled lab experiments: user feedback suggests that difficulties in articulating policies are not due to a poorly designed rule interface.

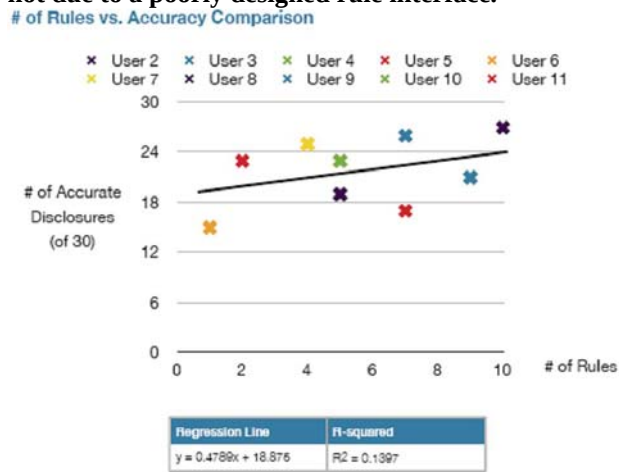

Figure 12. Controlled lab experiments: users reach a plateau, with little correlation between post-hoc accuracy and number of rules created

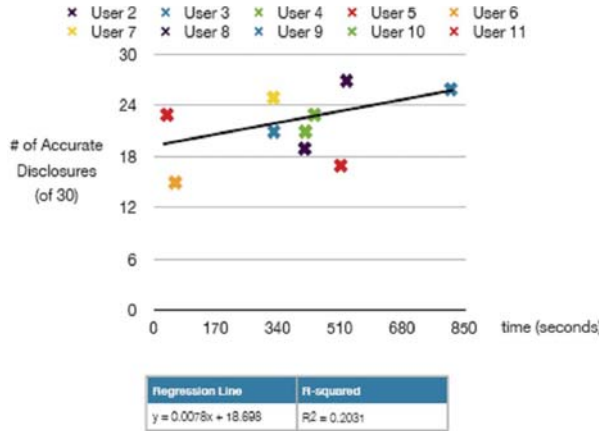

Figure 13. Controlled lab experiments: users reach a plateau, with little correlation between post-hoc accuracy and time spent defining and refining rules.

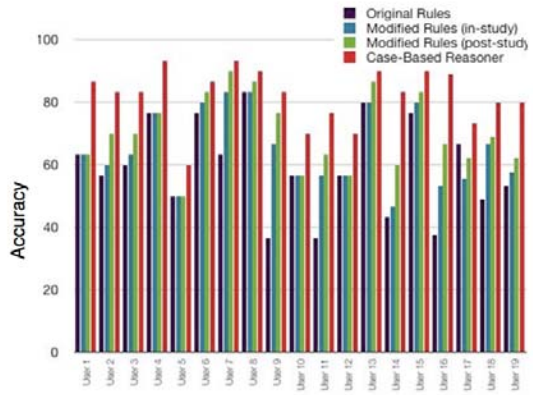

Figure 14. Controlled lab experiments: user feedback can help the system learn the user's privacy policy. This graph shows the performance of a person's original rules, modified rules, modified rules applied to all scenarios, and a version where user rules are refined with a case-based reasoner.

7. Field Studies

In Spring 2007, we shifted from laboratory studies to studies "in the wild," deploying a version of PEOPLEFINDER with three groups of target users. Each target user was asked to invite members of their social network and set up rules so that others could query their locations. The three groups of target users included (1) fifteen members of our research team, (2) a group of seven MBA students, and (3) a group of six people involved in organizing buggy races during Spring Carnival week at Carnegie Mellon. With the requesting users they invited, this amounted to a total of over 60 active users.

The pilot with members of our team spanned a total of six weeks. The pilot with MBA students lasted two weeks and the pilot with the Spring Carnival organizers spanned a total of nine days. Usage of the system was rather uneven with some target users having as many as 25 or more requesting users in their list of contacts and others having as few as one or two. For this reason, we limit the results presented in this section to the set of 12 most active target users (and their fairly large social networks), as measured by the number of daily requests submitted for their locations and amount of feedback provided through the auditing interface. This includes four members of our research team, two MBA students and all six Carnival users. Collectively, these target users were the subject of 1,314 location queries.

Overall the accuracy of the rules defined by the 12 most active users in these 3 pilot studies, as measured by the feedback they provided when auditing their logs, which was generally done once per day, was 79% (see Figure 15). This percentage is significantly higher than the 65% accuracy measured in laboratory experiments involving our PEOPLEFINDER mockup (see Section 6).

We believe that this difference can be attributed to five factors. First, we believe our participants were more careful in defining their rules, as they knew they were going to be used to process actual queries from friends and colleagues. Second, we believe that several improvements in the design of our system played a significant role in helping users define more accurate policies. In particular, this includes the introduction of functionality that lets users see detailed information about the context of each query and get explanations that identify the particular rules behind each disclosure/non-disclosure decision. Third, there were a significantly larger number of queries per user in the field trials than in our laboratory experiments, with over 100 queries per user versus 30 to 45 scenarios for users of our mockup application. This difference may also have contributed to the increase in accuracy. Fourth, the scenarios we chose for the laboratory study tended to examine situations where people might not want to disclose location information, but in real life these situations may not happen as frequently. Fifth, it is highly likely that participants in our lab studies simply did not have enough context to determine whether they wanted a disclosure or not, as our scenarios put them in hypothetical situations, whereas our field trials put them in actual ones. As suggested by Lederer et al. [20] and Consolvo et al. [5], "who is inquiring about one's location" is often the strongest factor in determining disclosures, but it is not sufficient in covering all possible situations. We believe that participants in our field trials had a better understanding of their personal context and situations in which they would want to share their location, thus leading to better accuracy.

While these results are encouraging, post-hoc experiments conducted using a random forest classifier [11] to refine a user's rules based on his or her feedback show that machine learning techniques offer the promise of higher accuracy levels than rules defined by users on their own. (see Figure 15). The challenge with such machine learning techniques is that they are usually configured as black boxes that take over, significantly restricting the ability of users to understand the rules that have been learned let alone tweak them. A more effective way to deploy these

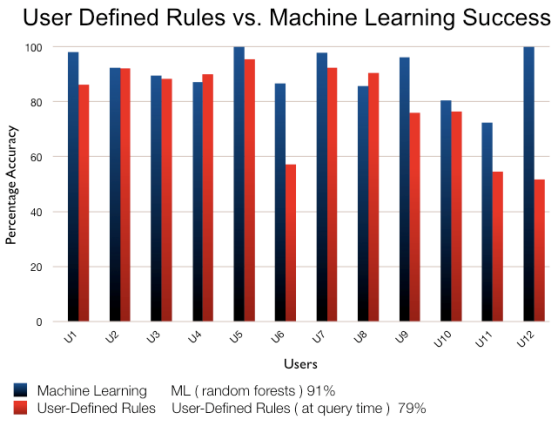

Figure 15. Field studies: accuracy for 12 most active target-users from 3 field pilots involving over 60 users. A random forest classifier shows promise in helping improve the accuracy of user-defined policies.

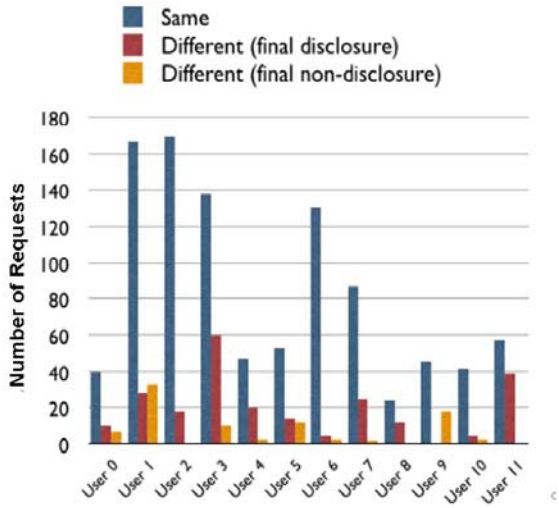

Figure 16. Field Studies: policy evolution during the pilot–12 most active target users. This compares the rules users originally defined with those they ended up with at the end of the pilot. "Different (final disclosure)" denotes requests that would originally have been denied but that eventually were allowed, whereas "Different (final non-disclosure)" denotes the opposite, and "Same" denotes no change.

techniques is in the form of suggestions for rule modifications that users can either accept or reject.

Figure 16 provides a more detailed analysis of how user policies evolve over time and suggests that users tend to initially err on the safe side as they define their policies. As they become more comfortable with the application and the way in which it is used by their acquaintances, they refine their policies and start allowing requests that initially would have been denied. Specifically, Figure 16 compares disclosure / non-disclosure decisions made by the user's final rules with those the user had originally defined. While the majority of requests resulted in the same decision ("same"), most of the decisions that are processed differently involve changes from non-disclosure decisions to disclosure decisions ("Different: Final Disclosure"). This was the case for 10 out of the 12 most active users.

8. Discussion

In this section, we take a step back and position our findings with respect to larger open issues the research community faces. Specifically, we examine three issues.

The first issue is helping people specify policies better. As noted in Section 7, there was a fairly large difference in accuracy between our lab studies (with user-specified rules achieving 65% accuracy) and our field trials (79% accuracy). Furthermore, it still takes a fair amount of time to specify policies up front, roughly five to eight minutes. Finally, as noted in Figure 16, people's policies seem to change over time. These results suggest that, beyond initial rule specification, there are still many opportunities for helping people refine and manage their policies over time. One possible approach would be to have basic patterns that people can easily choose from. For example, a "work" pattern might allow everyone tagged as a co-worker or boss to see one's location only while at the workplace, while a "family" pattern might allow everyone tagged as a family member to see one's location always.

The second issue is better ways of combining formal static mechanisms with dynamic social processes for managing location privacy. Our work in this paper represents one way of helping people manage their location privacy, focusing primarily on helping people craft better policies, providing adequate feedback, and examining whether machine learning techniques can help with policies. It is worth noting, however, that achieving effective location privacy in practice will likely require more than simply having effective policies. As Palen and Dourish argue, privacy cannot be managed simply by static enforcement of rules, but is instead a dynamic process of managing boundaries [21]. As such, effective location privacy

may require a combination of effective controls and feedback, coupled with social mechanisms such as plausible deniability and social translucency. Precisely what combination of formal static mechanisms and dynamic social processes are needed is still an open question.

The third issue looks at whether privacy should be managed primarily as an optimistic process or a pessimistic one. This might also be thought of as using a blacklist approach (optimistic approach where information is disclosed unless the user has specified otherwise) versus a whitelist approach (pessimistic approach where requests are by default denied – unless the user has specified otherwise). In our work, we opted for a whitelist approach, where people could specify rules that would allow individuals to see their location. This approach is consistent with the one generally taken by the security community. Our assumptions were that people would be reluctant to use a system that would make it too easy for anyone to see their location. Thus, we incorporated rules that would govern conditions in which the information would be shared with others. Interestingly, however, results from our field trials suggest that people are likely to relax their rules over time. We believe this may occur as people gain a better understanding of the capabilities and limitations of the system, see more value in letting others see their location, and see that others are not asking for their location as often or as intrusively as initially feared.

In contrast, Grudin and Horvitz [7] argue that a blacklist approach may be simpler to manage. They claim that with enough basic feedback, most people are unlikely to abuse the privacy of others. In cases where one's privacy is violated, an individual could then blacklist the offender. In this approach, people would generally be more willing to make their information available and add more restrictive policies only if and when there are abuses.

There are potential pros and cons to each of these approaches, regarding initial setup costs, correctness, comfort level, and overall utility. However, which of these approaches is better for location privacy, or how to combine the best aspects of these two approaches, is still an open question.

9. Concluding Remarks and Future Work

In this article, we presented our work on PEOPLEFINDER, an application that enables cell phone and laptop users to selectively share their locations with others. Our main objective has been to better understand people's attitudes and behaviors towards privacy with respect to one pervasive computing application, and to develop technologies and user interfaces that help users specify privacy preferences.

We conducted a laboratory study with 19 subjects as well as three field trials involving a total of over 60 participants. One interesting finding is that people have a hard time articulating effective privacy preferences. Functionality that increases user awareness of how the application is used and assists users as they audit queries (e.g. through explanation and access to detailed information about the context of each query) seems to help users define more accurate policies. Early results also indicate that machine learning techniques can help further improve accuracy. As part of our ongoing research, we are developing techniques that use machine learning to provide suggestions to users on how to refine their policies.

Another interesting finding is that people tend to be conservative about disclosures at first, but tend to relax their policies over time as they become more comfortable with PEOPLEFINDER and with how others are using it to find their location. This finding suggests that systems should help people stay in their comfort zones while also helping them evolve their policies over time.

Currently, we are continuing our work with PEOPLEFINDER, developing visualizations that can help people specify policies as well as see how their personal information is being accessed. We are also developing more sophisticated dialogues and explanations, to help people better understand the behaviors resulting from their policies and help them more effectively refine these policies. Finally, we are preparing for a larger study by making PEOPLEFINDER available as a FaceBook application, which we hope can help overcome critical mass issues, foster wider adoption, and enable larger-scale studies to be conducted.

10. Acknowledgements

This work is supported by NSF Cyber Trust grant CNS-0627513, NSF grant CNS-0433540, and ARO research grant DAAD19-02-1-0389 to Carnegie Mellon University's CyLab. Additional support has been provided by FranceTelecom, Nokia, IBM and Microsoft, the latter through the Center for Computational Thinking. PEOPLEFINDER's WiFi-based location tracking functionality runs on top of technology developed by Skyhook Wireless.

The authors would like to thank all the other members of Carnegie Mellon University's project on "User-Controllable Security and Privacy for Pervasive Computing" for their help designing and evaluating the PEOPLEFINDER application, including Lujo Bauer, Bruce McLaren, Mike Reiter, Jacob Albertson, Paul Drielsma, Jason Cornwell, David Hacker, Gary Hsieh, Jialiu Lin, Justin Pincar, Rob Reeder, Alberto Sardinha, Karen Tang, Janice Tsai, Kami Vaniea, Michael Weber, Wei Zhiqiang, and Yue Zhang.

References

1. Aha, D.W., Kibler, D. and Albert, M.K., Instance-based Learning Algorithms", in *Machine Learning*, Vol. 6, 37-66, 1991.
2. Barkhuus, L. Privacy in Location-Based Services, Concern vs. Coolness. In Proceedings of *Workshop paper in Mobile HCI 2004 workshop: Location System Privacy and Control*. Glasgow, UK 2004.
3. Beresford, A.R. and F. Stajano, Location Privacy in Pervasive Computing. *IEEE Pervasive Computing* 2003. **2**(1): p. 46–55.
4. Canny, J. and T. Duan. Protecting User Data in Ubiquitous Computing Environments: Towards Trustworthy Environments. In Proceedings of *Privacy-Enhancing Technologies (PET)*. Toronto 2004.
5. Cornwell, J., I. Fette, G. Hsieh, M. Prabaker, J. Rao, K. Tang, K. Vaniea, L. Bauer, L. Cranor, J. Hong, B. McLaren, M. Reiter, and N. Sadeh. User-Controllable Security and Privacy for Pervasive Computing. In Proceedings of *The 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile 2007)* 2007.
6. Consolvo, S., I. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. Location Disclosure to Social Relations: Why, When, & What People Want to Share. In Proceedings of *CHI 2005, Conference on Human Factors in Computing Systems*: ACM Press. pp. 82–90 2005.
7. Gruteser, M. and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In Proceedings of *The First International Conference on Mobile Systems, Applications, and Services (MobiSys 2002)* 2002.
8. Grudin, J. and E. Horvitz, Presenting choices in context: approaches to information sharing. 2003: Workshop on Ubicomp communities: Privacy as Boundary Negotiation. http://guir.berkeley.edu/pubs/ubicomp2003/privacyworkshop/papers.htm
9. Harper, R.H. Why People Do and Don't Wear Active Badges: A Case Study. In Proceedings of *Computer Supported Cooperative Work (CSCW96)*. pp. 297-318 1996.
10. Hightower, J. and G. Borriello, Location Systems for Ubiquitous Computing. *IEEE Computer* 2001: p. 57-66.
11. Ho, T.K. Random Decision Forest. In Proceedings of *The 3rd Int'l Conf. on Document Analysis and Recognition*. Montreal, Canada. pp. 278-282 1995.
12. Hong, J.I., *An Architecture for Privacy-Sensitive Ubiquitous Computing*, University of California at Berkeley, Berkeley, 2005.
13. Hong, J.I. and J.A. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In Proceedings of *The Second International Conference on Mobile Systems, Applications, and Services*. Boston, MA. pp. 177-189 2004.
14. Hsieh, G., K.P. Tang, W.Y. Low, and J.I. Hong. Field Deployment of IMBuddy: A Study of Privacy Control and Feedback Mechanisms for Contextual IM. In Proceedings of *9th International Conference on Ubiquitous Computing (Ubicomp 2007)* 2007.
15. Kaasinen, E., User Needs for Location-aware Mobile Services. *Personal and Ubiquitous Computing* 2003. **7**(1): p. 70-79.
16. Khalil, A. and K. Connelly. Context-aware Telephony: Privacy Preferences and Sharing Patterns. In Proceedings of *Computer Supported Collaborative Work (CSCW 2006)*.
17. Krumm, J. Inference Attacks on Location Tracks. In Proceedings of *Fifth International Conference on Pervasive Computing (Pervasive 2007)*. Toronto, Ontario, Canada, May 13-16, 2007 2007.
18. Krumm, J. A Survey of Computational Location Privacy. In Proceedings of *9th International Conference on Ubiquitous Computing (Ubicomp 2007), Workshop on Privacy*. Innsbruck, Austria, May 13-16, 2007 2007.
19. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, and H. J., J., Potter, F., Tabert, J., Powledge, P., Borriello, G. and Schilit, B.N. Place Lab: Device Positioning Using Radio Beacons in the Wild. In Proceedings of *International Conference on Pervasive Computing (Pervasive 2005)*. pp. To Appear 2005.

20. Lederer, S., J. Mankoff, and A.K. Dey. Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. In Proceedings of *Extended Abstracts of CHI 2003, ACM Conference on Human Factors in Computing Systems*. Fort Lauderdale, FL. pp. 724-725 2003.
21. Palen, L. and P. Dourish, Unpacking "Privacy" for a Networked World. *CHI Letters (Human Factors in Computing Systems: CHI 2003)*, 2003. **5**(1): p. 129-136.
22. Patil, S. and J. Lai. Who gets to know what when: configuring privacy permissions in an awareness application. In Proceedings of *The SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*. pp. 101-110 2005.
23. Priyantha, N.B., A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In Proceedings of *MobiCom 2000: The Sixth Annual International Conference on Mobile Computing and Networking*. Boston, Massachusetts: ACM Press. pp. 32-43 2000.
24. Rastogi, V., E. Walbourne, N. Khoussainova, R. Kriplean, M. Balazinska, G. Borriello, T. Kohno, and D. Suciu. Expressing Privacy Policies Using Authorization Views. In Proceedings of *9th International Conference on Ubiquitous Computing (Workshop on Privacy)*. Innsbruck, Austria, May 13-16, 2007 2007.
25. Sadeh, N., F. Gandon, and O.B. Kwon, Ambient Intelligence: The MyCampus Experience, in *Ambient Intelligence and Pervasive Computing*, T.V.a.W. Pedrycz, Editor. ArTech House, 2006.
26. Sohn, T., A. Varshavsky, A. LaMarca, M.Y. Chen, T. Choudhury, I. Smith, S. Consolvo, and W. Griswold. Mobility Detection Using Everyday GSM Traces. In Proceedings of *9th International Conference on Ubiquitous Computing (Ubicomp 2007)*. Irvine, CA 2006.
27. Tang, K.P., P. Keyani, J. Fogarty, and J.I. Hong. Putting people in their place: an anonymous and privacy-sensitive approach to collecting sensed data in location-based applications. In Proceedings of *Conference on Human Factors in Computing Systems*. Montréal, Québec, Canada: ACM Press, New York, NY. pp. 93-102 2006. http://doi.acm.org/10.1145/1124772.1124788
28. Want, R., A. Hopper, V. Falcão, and J. Gibbons, The Active Badge Location System. *ACM Transactions on Information Systems* 1992. **10**(1): p. 91-102.
29. Wireless, S. http://www.skyhookwireless.com