Super-Ego: A Framework for Privacy-Sensitive Bounded Context-Awareness

Eran Toch
Department of Industrial Engineering
Tel Aviv University
Ramat Aviv, Tel Aviv 69978, Israel
erant@post.tau.ac.il

ABSTRACT

Context-awareness enables applications to better streamline and personalize their service according to the current situation of the user. However, the user's information used by context-aware applications, such as the user's current location, is inherently private and sensitive. Using this information without proper control by the user can lead to privacy risks and might harm the trust users have in the context-aware application. To address this tradeoff between the effectiveness and privacy, we present Super-Ego, a framework for at-hoc management of access to location information in ubiquitous environment. Using this framework, we model and evaluate different decision strategies for managing mobile application's access to location context. The strategies we test are based on automatic algorithms that use knowledge about historical disclosure of locations by large number of users, with the optional delegation of some of the decisions to the user. We evaluate the system empirically, using people's detailed location trails from public resources, augmented with simulated data about sharing behavior. Our results reflect on an interesting tradeoff between automation and accuracy, which can enable the design of efficient and usable approaches to privacy-sensitive context-aware applications.

Keywords

Context-awareness, autonomous systems, privacy, usability

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and protection; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces

1. INTRODUCTION

Ensuring users' privacy is becoming a major challenge in context-aware applications. As mobile applications increasingly rely on automatic context sensing to simplify and personalize services to users, users may find it difficult to trust the process in which services collect and use their context information. Users need to know that their information is collected and used in a way which is consistent with their expectations. Otherwise, their information might be taken out of its intended context, and used in ways which may harm their privacy. This is becoming a challenge for designers of autonomous context-based systems due to two dramatic trends. The first trend is the huge advanced of mobile technology that simplifies the way application collect diverse context information, including exact physical location, proximity to other users, interaction with other users, calendar information and so forth. The second trend is the wide adoption of social networks, which increases the possible use of context information. Context information can now be reported to friends, family, co-workers and other social relations, complicating privacy risks and making them tangible to users.

How can we design self-managed systems that protect users' privacy? Current theories highlight the inherent challenges in protecting privacy by self-managed systems, and in particular in systems that are used for sharing information between users. Helen Nissenbaum's contextual integrity theory explains why transmitting information about a person from the original context to a new context can lead to privacy risks [13]. For example, information about the user's location can be safely shared with work colleagues during work hours, but not with the same people during the night. The challenges users face when managing the context of their information is becoming increasingly difficult, when it is shared in expanding social contexts. The control users have over how their information is collected and used is crucial for their sense of privacy and identity management [14]. By definition, self-managing systems strive to act as independently as possible, which can lead to compromised sense of privacy if systems use information in contexts that the user did not anticipate and cannot control.

In this paper we address the tension between autonomy and privacy by suggesting a limited approach to context-awareness. We that using the "Super-Ego" framework bounded context-awareness for location context information. The framework controls the flow of context information from the mobile phone to context-aware applications. As in Sigmund Freud's structural model, where the super-ego plays the critical and moralizing role in our mental life, our framework plays a similar part to self-managed context-aware applications. When a context-aware application requires a location context information, it requests the location from Suepr-Ego, which uses a mixture of automatic and manual decision making strategies to decide whether to accept or reject the request.

The automatic decision algorithms rely on existing information about the disclosure of past location contexts. Several empirical works show that users consistently discriminate between location context disclosure, and that some instances are considered more private than others [11, 10, 3, 4]. Empirical works also proved

Casemans '11, Sep 18, 2011, Beijing, China.

that there is strong commonality between different users when it comes to decisions regarding location disclosure [2, 16]. Given a set of location disclosure decisions by the general population, Super-Ego determines whether a decision can be decided automatically, or should be decided by the user. For example, let us imagine a location-based dating service that uses the location of the user when other users wish to see if there are possible romantic partners nearby. The dating application, which is installed on the user's smart-phone, would request the location from the Super-Ego framework. The framework would learn whether the decisions of the general population and would either provide the current location to the application, deny it, or let the user decided if no confident decision can be made automatically. For isntance, if the user is currently sitting in a coffee shop, a place with high probability of being shared, then the framework would release it automatically. On the other hand, if the user is currently at home, Super-Ego would delegate the decision to the user.

Super-Ego defines quantitatively the boundaries of self-manageability with respect to the desired accuracy of the decision process and the required user involvement. We use the framework to develop and evaluate a model for location context disclosure using measures from information retrieval and human-computer interaction. We develop an evaluation methodology that takes into account the accuracy of our the decision strategy and the level of automation the strategy provides. We evaluate our approach using actual location data of 21 users collected over a 2 month period, made available by Microsoft Research [6]. The data is enhanced with simulated sharing preferences that mimic reported location sharing behavior in several research papers [16]. Our results show that semi-manual strategies exhibit an optimal accuracy/automation balance when both of these parameters are considered as significant.

To summarize, the contributions of the papers are threefold: we propose an architecture for privacy-sensitive context-aware computing, we develop models for location context management based on historical disclosure preferences, and we analyze the properties of these models with respect to accuracy, automation and overall efficiency.

2. PRIVACY AND CONTEXT-AWARENESS

In existing mobile operating systems, controlling location context collection is very limited. In the Google Android operating system, mobile applications request permission for accessing operating system resources, including context information such as exact location in different levels of granularity [1]. When the user installs a new application, the list of requested permissions is presented, and the user can choose to accept or reject the installation. In the Apple iOS operating system, deployed on iPhones, at the first time that the application requests the location, the system presents a modal dialog that asks the user whether to provide the location. If the answer is positive, the location is released to the application, and the user's decision is set as default for that particular application. In both operating systems, and in most mobile operating systems in general, location disclosure decisions are done at the application level, either allowing the application full access to all future locations or to none.

In these "all or nothing" approaches to location context utilization, users cannot differentiate between private locations, which they do now wish to disclose, and locations they do wish to disclose. However, empirical evidence shows that users have detailed preferences regarding disclosure of specific locations. The willingness of users to share their location depends the specific identity of the person receiving the location [3], the activity of the user in the location [10], the time and place [4], and the properties of the location, e.g., the variety of people that visits the location [16]. Therefore, we believe that frameworks for managing location context information should be able to provide users with fine-grained and usable control over individual locations.

Context-aware applications use context information about the state of people, places, and objects relevant to users and their activities to adapt the applications' behavior [8]. As information about the state of the users is inherently private, the tension between privacy and context-awareness is an ever challenging research question [**?**]. It has been addressed by using technologies such as rule-based policies [9, 4], conflict specification with automatic resolving [17], and setting context roles for expected contexts [7]. Location obfuscation was also suggested to be used to restrict context awareness, by limiting the level of detail given about a location [**?**]. Our work complements these research efforts by developing a context-aware framework that manages disclosure of location context data. The work takes a different approach, relying on at-hoc decision making regarding the disclosure of the location, on the basis of historical sharing data from the general population with optional interference by the user.

3. THE SUPER-EGO FRAMEWORK

In this section we explain in details how Super-Ego implements fine-grained automatic location context control. We explain how the framework is embedded within contemporary mobile operating systems and explain the concept of decision strategies and the research questions they pose.

3.1 Architecture

The framework we present offers a simple architecture that insures privacy-sensitive context-awareness. Current mobile operating systems provide an API (Application Programming Interface) that is used by mobile application to access context information and other operating system resources. The Super-Ego framework is positioned between the original operating system API and the mobile application. In our approach, mobile applications access context using the framework, which decides whether to grant access to the context information. As Figure 1 depicts, when a mobile application request a context information from the operating system API, the request first go to the Super-Ego framework, that decides whether to release the context information. If the request is granted, then the mobile application can function on the context. Otherwise, the mobile application should be able to handle the rejection in a user-friendly way.

Decisions on whether to release the context information are based on a **context model** that is maintained by the framework. The context model includes the set of earlier locations requested from the mobile operating system API, as well as historical knowledge of the disclosure decisions regarding the current location by the current user and by all other users of Super-Ego. The general knowledge is kept in a centralized server, and contains information about the ratio in which the current location was disclosed by Super-Ego users. For the sake of privacy, the actual decisions of individual users are not stored.

The heart of the framework is a decision engine that computes the response to the location requests. The response regarding a particular location request can be one of three outcomes: **disclose** - to accept the request, **deny** - to reject the request, or **manual** - to let the user decide. If the response is disclosed, then the location is returned to the mobile application. If the response is to deny the request, then the function throws an exception that the calling application need to handle. If the response is manual, then the user is presented with a user interface that asks whether to release the lo-

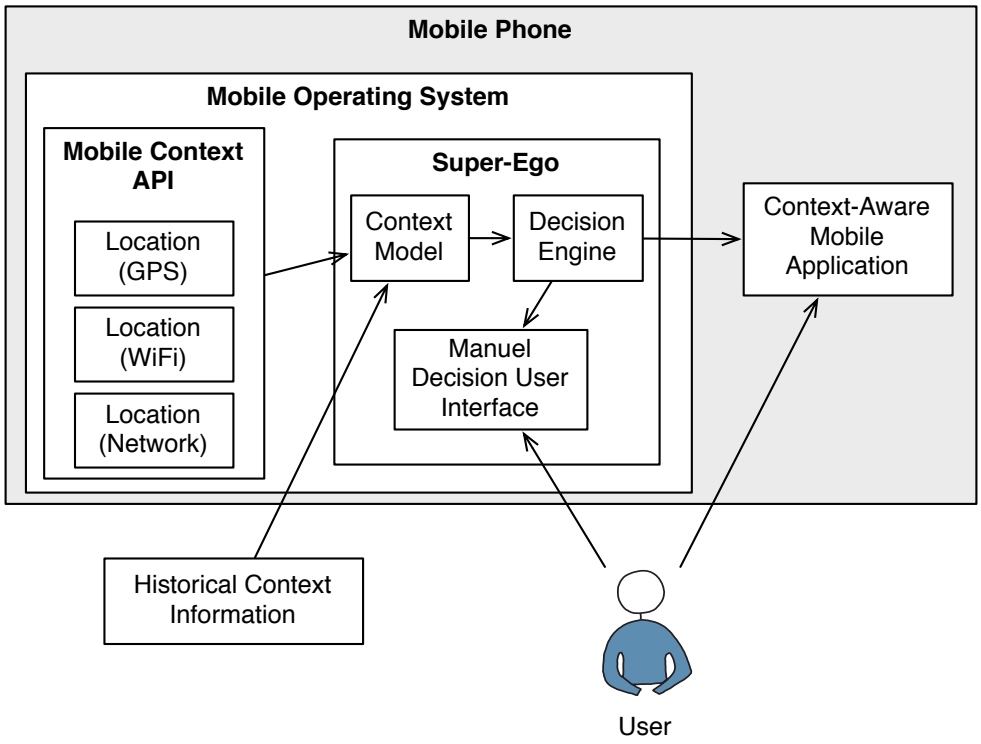

Figure 1: The architecture of the Super-Ego framework. Super-Ego is embedded within the mobile operating system (e.g., Android, iOS, or Windows Mobile 7.) A context-aware mobile application is the client of Super-Ego, and accesses it when a location context is required. The Super-Ego framework uses two external resources when deciding on location disclosure: data about historical context information and the user herself.

cation to the current application. From this point, Super-Ego would implement whatever the user had decided.

Super-Ego was designed to adhere to privacy-by-architecture principles, promising that mobile applications cannot achieve undesired context information due to malicious or buggy infrastructure. Therefore, in our theoretical architecture, Super-Ego is embedded within the operating system, and used as the sole methods for accessing location contexts by applications. Super-Ego exhibits several other properties:

1. Variable manual control: allowing manual intervention and decision in certain cases, when the certainty of automatic decision is low.

2. General and historical knowledge: basing the context release decisions on past history and global context models, due to evidence showing strong diversion to the mean when it comes to information sharing preferences.

3. Configurable automation: Super-Ego can be configured to require different levels of user involvement.

Super-Ego was implemented on Android 2.3 mobile operating system. While not impossible, embedding the framework within the operating system takes great effort, and therefore it was implemented as a Java library that can be used in mobile applications that requires access to location context. The library wraps most of the native location API and provide access to them through a set of methods that first call the decision engine.

3.2 Decision Strategies

The main focus of our work is to develop and evaluate different strategies for deciding the release of location context information to mobile applications. We are interested in an open framework that can exhibit different strategies for both engineering and research standpoints. From the engineering standpoint, allowing configurable strategies can be used to personalized specific strategies to specific users or applications. From the research standpoint, configurable strategies make it easier to formally model decision algorithms and to systematically evaluate them.

While strategies can be very complex, in this work we focus on simple strategies that can be configured using a simple set of parameters. The simplicity of the strategies described in this paper allows us to compare them in a straightforward manner, revealing relations between the strategy performance and its specification. A strategy is basically specified using two parameters: manual threshold and disclose threshold. Below the manual threshold, the decision engine would deny the request. Above the manual threshold and below the disclose threshold, the engine would send the request to manual intervention, and above the disclose threshold, the engine would disclose the location. Strategies differ by the two parameters, as well as by methods for dynamically setting the disclose threshold.

The use of strategies in a framework that allows both manual and automatic decision making, raises several questions which are critical to the understanding of using privacy-sensitive context-awareness frameworks. Specifically, we are interested in evaluating the amount of manual intervention required in a specific strategy, and its impact on the accuracy of the decision engine. We ask several research questions, which are answered in Section 5.

1. What is the effectiveness of using general historical information about location disclosure decisions?

2. What is the impact of manual decision delegation on the automation, accuracy and overall performance of decision strategies?

3.3 Model for Bounded Awareness

Decisions in Super-Ego are taken for each instance of location context requested by a mobile application. At each request, the decision engine operates on a given location context, l_k. Each location is basically a pair of longitude/latitude coordinate. In order to perform a decision, the decision engine uses a **context model**, $\mathcal{M}$, which documents the privacy preferences for all known locations from all known users of the framework. Prior decisions are represented by the ration, $r \in [0, 1]$, the average of the scores for sharing the location and $sd(r)$, a number that represents the standard deviation of that ratio. For example, if a location l_k was considered by sharing by n users, each of them giving a score between 0 to 1 for comfort in sharing the location, then r(ratio) is the mean of those scores and $sd(r)$ is the standard deviations of the scores.

We define the context model as a mapping function $\mathcal{M} : L \rightarrow r \times sd(r)$ that assigns a disclosure ratio and standard deviation to each recorded location. We identify locations using an approximation of 25 meters, so when the decision engine judges a location l_k, we look at all locations $\mathcal{M}(l_k) = \{l_i \in dom\mathcal{M} | dis(l_i, l_k) < 25m\}$. We then look at the average of ratios and standard deviations for all locations in $\mathcal{M}(l_k)$.

The decision engine computes a decision based on a decision **strategy**, a construct that is used to configure the decision algorithm and to set the extent of manual intervention the decision algorithm will yield. As depicted in Figure 2, we represent the strategy as a set of two threshold values t_{manual} and $t_{disclose}$ on the range of the ratio$[0, 1]$. We define the decision engine as a function that takes a set of three elements: a context l_k, a context model $\mathcal{M}$ and a strategy $\mathcal{S}$ and returns a decision:

$$F : l_k \times \mathcal{M} \times \mathcal{S} \rightarrow \{disclose, deny, manual\}$$

The outcome of the function is set by a straightforward algorithm:

$$F(l_k, \mathcal{M}, \mathcal{S}) = \begin{cases} disclose & r > t_{disclose} \\ manual & t_{manual} > r > t_{disclose} \\ deny & r < t_{manual} \end{cases}$$

Let us exemplify the way the model of the decision engine works using the following scenario. A mobile application requests the location of the user, l_i. The historical ratio for that location is: $r = 0.9$ with standard deviation of $sd(r) = 0.081$. If the historical ratio is lower than t_{manual}, the location is disclosed, if the ratio is between t_{manual} and $t_{disclose}$, then the user is asked to weigh in on the decision, and if the ratio is higher than $t_{disclose}$ then the location is disclosed without any user intervention. For example, if $t_{disclose} = 0.8$, then the location will be disclosed. Setting the thresholds can yield dramatically different behavior, as we exemplify in Figure 2. If the strategy is set as: $\mathcal{S} = \{t_{manual} = 0, t_{disclose} = 1\}$ then the decision engine is fully manual, as all ratios will be above the manual threshold and below the disclose threshold. In that case, the outcome of our example scenario is manual. If the strategy is set as: $\mathcal{S} = \{t_{manual} = 0.5, t_{disclose} = 0.5\}$ then the decision is fully automated as all ratio values will be either smaller than t_{manual} or higher than $t_{disclose}$. In a semi-manual strategy, the threshold values will be set up is some distinctive way that would reflect the desired amount and nature of user intervention.

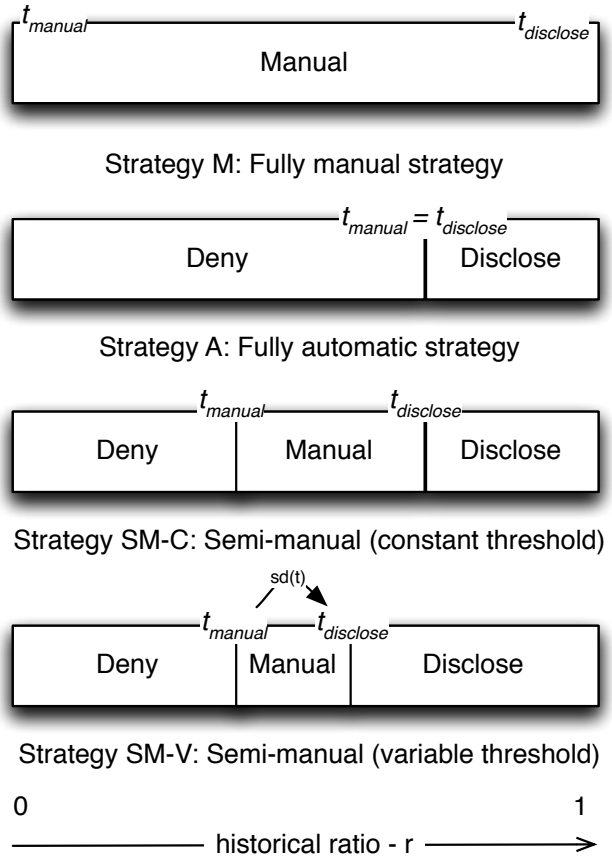

Figure 2: A depiction of four strategies: fully manual, fully automatic, semi-manual with constant threshold and semi-manual with variable threshold. Each strategy is specified using two threshold values: t_{manual} and $t_{disclose}$, which are used to judge location requests according to the ratio of historical decisions for the given location. In the diagram, each strategy is displayed with the thresholds visibly located where their values are set for that strategy. In the last strategy, semi-manual with variable threshold, the arrow between the threshold depict the fact that $t_{disclose}$ is set dynamically according to the standard deviation of the historical location data from the context model.

4. EVALUATION METHODOLOGY

In this section, we explain our empirical framework for assessing decision strategies. We first define measures for strategy assessment, and then follow by providing a detailed explanation about our experimental testbed.

4.1 Measures for Estimating Strategies

In order to measure the performance of decision strategies, we define a methodology that focuses on two aspects: accuracy and automation. This methodology will enable us to compare strategies and simulate how different strategies would impact the user interaction and the overall behavior of the system. The performance measure is a combination of these two aspects of a strategy. If a decision engine involves the user in every decision, it might get perfect results, but would compromise the usability of the application through excessive user burden. On the other hand, if an engine requires no manual intervention, its accuracy might be mediocre. The

objective of our evaluation methodology is to enable us to characterize how well a strategy fits in this tradeoff between accuracy and automation, and help us identify good strategies that balance these two important aspects.

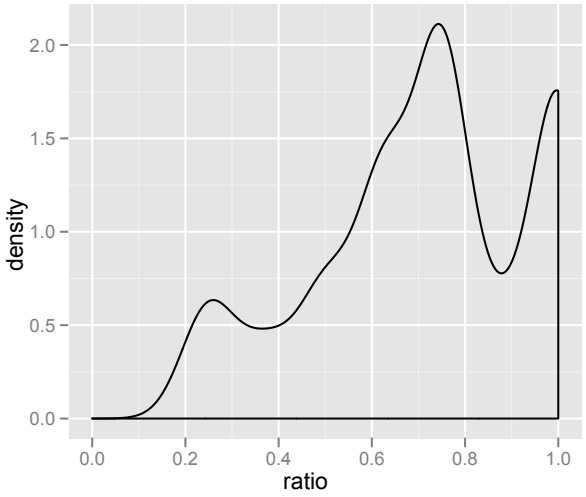

Figure 3: The density of the historical ratio for all simulated location context. The ratio has local peaks on the extreme values of the scale (1-2 and 10), and on values which are slightly above the average (6-8.)

We assume that we run a decision engine F on an identical set of locations $L = \{l_1, l_2, \cdots, l_n\}$ and an identical context model $\mathcal{M}$. Every request which is decided upon by the decision engine, can turn out to be a positive decision (disclose) or negative (denied). We can then judge the performance of the decision according to the actual decision made by the user. Information about these decisions was obtained, in our case, by a-priory simulation. If the user agrees with the decision engine, then the decision is called true. Otherwise, it is called false. We assume that the user is always satisfied with the result of a manual request, and therefore manual decisions are always considered true. Therefore, information retrieval categorization can be used on the output of the decision engine, resulting in four categories:

- True positive (tp) - the decision was to disclose, and the prediction was correct (i.e., the prior user decision agrees with the outcome of the decision engine.)
- False positives (fp) - the decision was to disclose, and the prediction was incorrect.
- True negatives (tn) - the decision was to deny, and the prediction was correct.
- False negatives (fn) - the decision was to deny, and the prediction was incorrect.

On the basis of the categorization of result satisfaction, we employ standard measurements from information retrieval, namely precision, recall and accuracy. We define the precision of a strategy as:

$$precision(S) = \frac{tp}{tp + fp}$$

and the recall of the strategy as

$$recall(S) = \frac{tp}{tp + fn}$$

A conservative decision engine that uses a high threshold (hypothetically returning few true positives and many false negatives) will have high precision but low recall. A liberal decision engine that uses a low threshold (hypothetically returning many false positives and few false negatives) will have a low precision but high recall. We evaluate the overall accuracy using the standard accuracy function used in information retrieval, giving equal weights to both measures:

$$accuracy(S) = \frac{tp + tn}{tp + tn + fp + fn}$$

In order to evaluate the user involvement for each of the strategies, we count the number of manual decisions and the number of automatic decisions. We define the user involvement of a given strategy as the ratio between automatic decisions and the overall number of decisions. We denote by a the number of decisions taken autonomically by the decision engine, and by m the number of decisions sent to the user. We define the automation measure as:

$$automation(S) = \frac{a}{m + a}$$

The two measures, accuracy and automation, reflect respectively how well an algorithm decides regarding a set of locations, and how much user intervention is required. To evaluate the overall performance of an algorithm, configured by a strategy, we developed a simple combined measure, which we call the combined score (or "combined" for short). We define the score as follows:

$$combined(S) = \alpha \cdot automation(S) + (1 - \alpha) \cdot accuracy(S)$$

The combined score is a sum of automation and accuracy, weighted by a coefficient $\alpha \in [0, 1]$ which sets the ratio between the two measures. For example, when $\alpha = 0$, a high score would be given to a strategy with high accuracy with no regard to automation. When $\alpha = 1$, the only meaningful measure would be automation, and when $\alpha = 0.5$, equal importance would be given to both measures.

4.2 Evaluated Strategies

In evaluating the strategies we had devised and implemented four strategies that represent different variations of strategy types, as depicted in Figure 2. We denote by $x \in [0, 1]$ a threshold that serves as a variable in the experiments. The following four strategies were evaluates:

- Manual strategy (**M**): all requests are decided as manual, such that $\mathcal{S} = \{t_{manual} = 0, t_{disclose} = 1\}$.
- Fully automatic strategy (**A**): all requests are decided automatically, such that $\mathcal{S} = \{t_{manual} = x, t_{disclose} = x\}$
- Semi-manual with constant threshold (**SM (C)**): in this strategy, $t_{manual} = x$, and $t_{disclose} = t_{manual} + \Delta$, where Δ is a constant. The constant was arbitrarily set up to the average standard deviation of all disclosure rates for all locations in the context model. In our experiments, $\Delta = 0.229$.
- Semi-manual with variable threshold (**SM (V)**): in this strategy, t_{manual} is a variable, but unlike the previous strategy, Δ is varied and changes between locations. It depends on the standard deviation of rates for that particular location, such that for a specific location l_i with a ratio r, $t_{manual} = x - sd(r)$ and $t_{disclose} = x + sd(r)$.

The logic behind choosing these particular strategies is as follows. The automatic and manual strategies were designed to understand the boundaries of the automation/accuracy tradeoff. The two strategies represent the baseline for comparison for both of the tradeoff extremes. The two semi-manual approaches were designed to search for a balance between accuracy and automation using different approaches for decision making.

4.3 Experimental Setup

The context model used in the evaluation is based on a data set of the GPS coordinates of 21 different users, provided by Microsoft Research [6]. The locations were recorded through a period of 2 months by a GPS device that sampled the location every 5 seconds. Overall, there were 3,082,900 location observations. On top of these locations, we had simulated disclosure ratios for a sample of the locations. The simulated disclosure ratio followed the properties reported in [16], which report relations between the history of place visitation (by the user and by all users) to the likelihood of sharing the location. In assigning sharing ratios for these locations, we boosted the ratio according to the empirical model, giving the number of times the user visited the location and the entropy of the location (a measure for the diversity of the location.) The average values of disclosure ratios are displayed in Figure 3. The mean ratio is 0.711 and the standard deviation is 0.22. Extreme points have local peaks, as some locations are almost never shared (and therefore the ratio for these location is 0) and other points are shared by several users.

When running the experiment, we had evaluated each location in the data set using each of the strategies. The decision for each of the locations was saved and compared against the original simulated decision of the user. The decision was then analyzed according to the methodology listed above.

5. RESULTS

The four strategies differ significantly with respect to their accuracy and automation. The following table summarize the accuracy and automation of the strategies. Values are means and values in parenthesis is the standard deviation.

Strategy Performance		
Strategy	Accuracy	Automation
A	0.594 (0.075)	1 (0)
M	1 (0)	0 (0)
SM (C)	0.685 (0.06)	0.819 (0.131)
SM (V)	0.64 (0.052)	0.919 (0.089)

The difference between the strategies, when it comes to automation and accuracy, is significant. Results were obtained In a two-sample independent t-tests with unequal variances, with $p < 0.001$ for every strategy pair. The test details for the comparison of accuracy between the two semi-manual strategies are: t = 3.3219, df = 73.834 and the 95% confidence interval are 0.0179 and 0.0716.

Our second set of results, presented in Figure 4, investigates the accuracy of each of the different strategies, for different threshold t_{manual} values. As we assume that all manual decisions are correct (true positives and true negatives), it is not surprising that the manual strategy (M) has perfect accuracy. The fully automatic strategy (A) is producing reasonable results, when framed as an information retrieval problem and compared to similar problems. It exhibits a with maximal accuracy of 0.652, with precision of 0.714 and recall of 0.685. However, the automatic strategy is outperformed by the two semi-automatic approaches, as they are asking the user to manually intervene on some of the location contexts.

For the semi-manual approaches, the best accuracy is achieved with a threshold of 0.7, for both semi-manual approaches. At those points, the semi-manual with constant threshold, provides an accuracy of 0.74 with precision of 0.88 and recall of 0.722 and the semi-manual with variable threshold provides an accuracy of 0.7 with precision of 0.79 and recall of 0.7. This boost in accuracy can be explained by turning the decision to manual intervention, which results in true positives and true negatives, thus increasing both the recall and precision of the strategy. Furthermore, as the decision algorithm outputs location contexts with medium grades as manual results, they receive a boots exactly where automatic algorithms will risk outputting a decision that will be either false positive or false negative.

While manual and semi-manual approaches are outperforming automatic approaches, they have a cost in decreased level of automation. Figure 5 presents the levels of automation for each of the strategies. Naturally, the fully manual approach has no automation. On the other hand, the fully automatic strategy has maximal automation regardless of the threshold. The semi-manual strategies exhibit variable levels of automation, based on the threshold, as the decisions that result in manual intervention (and therefore decrease automation) are based on the threshold. The highest levels of automation are perceived in either a very low threshold (where most requests are denied) or a very high threshold (where most requests are allowed.) The lowest levels of automation and the highest levels of accuracy for the same threshold levels (0.675 0.825). At those points the constant strategy delegates 4 our of 10 requests to the user, and the variable approach delegates 2 out of 10 requests. This phenomena has several reasons. First, in these ratio values, there is a lot of variability, as can be seen in Figure 3. Second, the variability increases the chances that a location will be sent to manual rather than being denied. In result, the efficiency of the strategy increases, on the expense of its automation.

We now consider how the strategies differ with respect to the tradeoff between automation and accuracy. Figure 6 shows the combined score for all four strategies, with a variable threshold and five sub-diagrams according to a variable α value. The combined score is configured by the α coefficient, where $\alpha = 1$ gives all the weight on accuracy and $\alpha = 0$ gives all weight to automation. The sub-diagrams in Figure 6 are ordered from left to right according to the α values, ranging from giving full weight to automation (on the left) and full weight to accuracy (on the right). Here, we see a generalization of the results shown in Figures 4 and 5, expressing an accuracy-automation tradeoff. The manual strategy has constant accuracy, and therefore it is dependent only on the α value, having the worst combined score when $\alpha = 0$ and the best combined score when $\alpha = 1$. Similarly, the combined score of the automatic strategy is linearly decreasing as it is dependent on the proportion of accuracy in the overall score. The tradeoff is particularly telling when it comes to the semi-manual approaches. The semi-manual strategy with constant threshold trades automation with accuracy, and is less sensitive to the α value than the automatic approach. The variable approach outperforms the constant strategy when automation plays a meaningful role in the combined score ($\alpha < 0.7$), as it is relatively less dependent on manual input to provide accuracy.

In analyzing Figure 6, the tradeoff between automation and accuracy is very clear. In delegating tough decisions for manual intervention, algorithms can increase their accuracy and outperform fully automatic strategies. When automation is factored into the combined score, reflecting the need to minimize user burden, the score of manual and semi-manual strategies are decreasing. Approaches such as the semi-manual approach with variable threshold, which are more selective in the contexts they decide as "manual", are more robust when the need to minimize user involvement

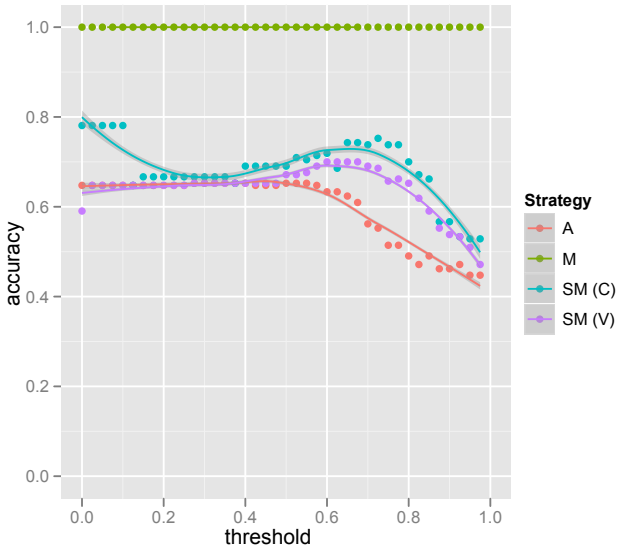

Figure 4: The accuracy versus a variable threshold value t_{manual} for each of the four strategies. Points represent data observations while curves shows moving average (with 95% confidence intervals.)

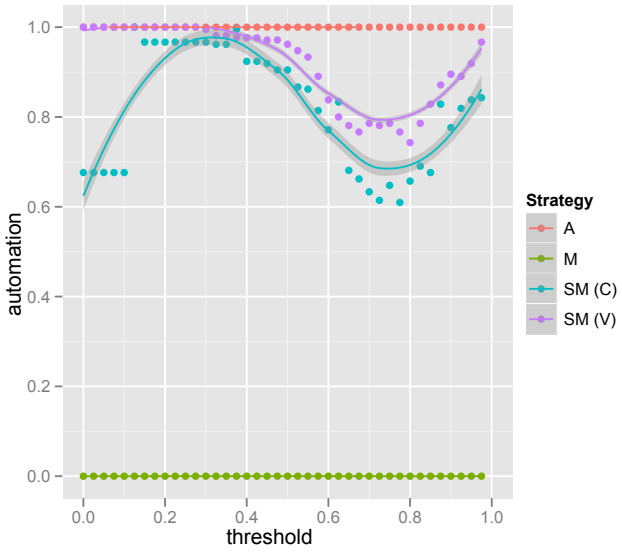

Figure 5: The automation versus the threshold value t_{manual} for each strategy. Points represent data observations while curves shows moving average (with 95% confidence intervals.) Note that the automatic strategy has constant automation level.

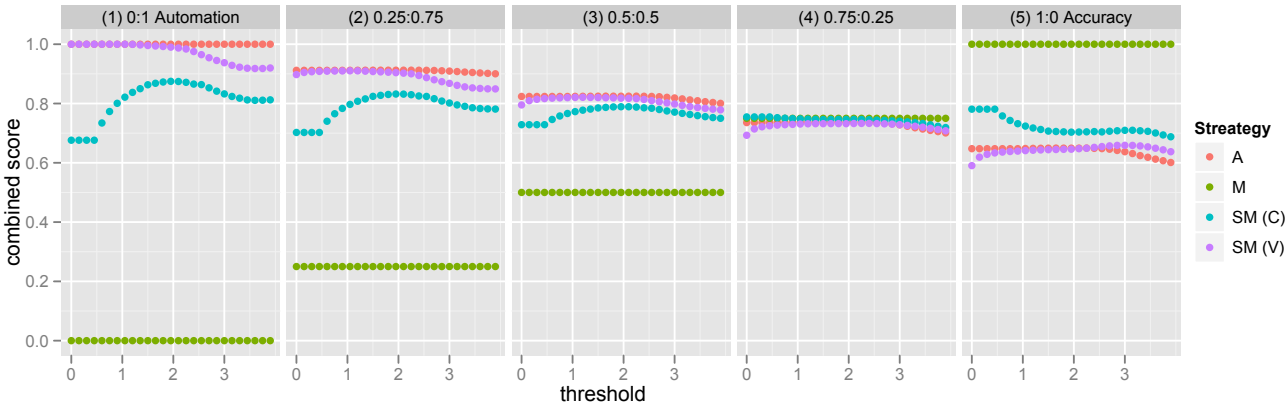

Figure 6: The combined score versus a variable threshold for each of the strategies, given according to 5 α values from 0 (on the left) to 1 (on the right). The combined score weights accuracy by α and weights automation by $1 - \alpha$, such that: $combined = \alpha \cdot automation + (1 - \alpha) \cdot accuracy$. Therefore the combined score starts from a combination that expresses just automation (on the left) all the way to a combination that expresses just accuracy (on the right). In each of the five α value settings, we show the combined score depending on a moving threshold.

is important.

6. DISCUSSION AND LIMITATIONS

In this section, we take an overview of our findings with respect to larger open issues the research community faces. Specifically, we examine three issues.

The first issue is managing the tradeoff between automation and accuracy in questions related to privacy. The research community and the media are well aware that one of the main problems in managing privacy is its substantial requests on the user's time and effort [4, 15, 5]. The Super-Ego framework manifests this tradeoff as part of its inherent mechanism, making it possible to configure the relations between automation and accuracy. The simple framework we propose for evaluation, presented in Section 4 can be used to systematically evaluate the tension between automation and accuracy in user studies and for different algorithms.

The second issue is better ways to personalize context disclosure policy according to the user's particular privacy profile. As the literature of privacy preferences repeatedly tell us, people have distinct privacy preferences that follow profiles such as those described by Alan Westin: privacy fundamentals, privacy pragmatists and unconcerned [12]. The model coefficients presented in Section 3.3 can be used to personalize the user experience for different categories of users. For example, by providing a less automated

strategy for users who are privacy fundamentals.

The third issue relates to using Super-Ego in solutions for managing context information using a combination of at-hoc decision strategies solutions and pre-defined rules. Several works have shown how pre-defined specifications of rules, conflict specifications and roles can effectively be employed in privacy-sensitive contextawareness [9, 4, 17, 7]. Combining rule-based decision models with at-hoc decision models, such as Super-Ego, can result in usable privacy management systems. In a combined approach, the user's known restrictions can be expressed directly using a rule-based interface. Unexpected situations, which are not well defined by rules, can be handled by Super-Ego, with its combination of automatic and manual decision processes.

The approach we present in this paper is limited in several ways. First, Super-Ego requires knowledge about location disclosure behavior from the general population. While wide-spread adoption of Super-Ego can eventually lead to create such a knowledge base, it is currently nonexistent. Moreover, in this paper we do not resolve potential privacy risks that stem from sharing historical location context disclosure decisions. The second limitation is the architecture of Super-Ego, which requires all context requests to go through a single filtering layer. This property can eventually limit the applicability of the approach. The third limitation is the equal weight we give to false positives and false negatives. In most scenarios the outcome of a wrongful disclosure of a private location can be considered more harmful than denying a mobile application of a location. Finally, we do not take into account in the decision process the different applications that request the information, and the different uses the applications might use the location context for.

7. CONCLUSIONS

In this paper, we present a method for preserving privacy in context-aware systems using at-hoc decision making. We had implemented and evaluated our approach for a particular type of context: locations sensed by the mobile device. We present an architecture in which a filtering layer, called Super-Ego, is placed between the mobile platform's operating system and mobile applications that require the user's context information. As mitigating the task of deciding about the the release of every location to the user will compromise the usability of the framework, we develop and evaluate a model for decision strategies that combine automated methods and manual intervention.

The evaluation of Super-Ego is based on simulated data sharing preferences, generated on the basis on actual location data gathered from by Microsoft Research of 21 users tracked for 2 months. The empirical evaluation portrayed the tradeoff between accuracy and automation with respect to different decision strategies, including manual, automatic, and various semi-manual approaches. While a fully automated approach delivers reasonable results (precision of 0.7 with recall of 0.8), they are less accurate than manual and semi-manual approaches. Our results show that having the user interfere in even a small part of the location contexts boosts the accuracy of the decision process. While user intervention reduces the automation of a given strategy, it is possible to quantify and adapt the strategy to the desired amount of user involvement.

The findings in this paper open several possibilities for future work. One possibility involves exploring additional dimensions of automatic decision making, based on learning algorithms. Algorithms that use machine learning and advanced location data sets to provide better predictions for managing contexts, improving the precision and recall of current approaches.The second possibility is investigating the impact of accuracy models that provide different weights to positive and negative errors. For example, studying the difference between models that are less tolerant to releasing unwanted locations than to blocking legitimate locations.

8. REFERENCES

[1] Android Developer Guide. Security and permissions. http://developer.android.com/guide/topics/security/security.html, June 2011.

[2] D. Anthony, D. Kotz, and T. Henderson. Privacy in location-aware computing environments. *IEEE Pervasive Computing*, 6(4):64–72, 2007.

[3] L. Barkhuus, B. Brown, M. Bell, S. Sherwood, M. Hall, and M. Chalmers. From awareness to repartee: sharing location within social groups. In *CHI '08*, pages 497–506, 2008.

[4] M. Benisch, P. Kelley, N. Sadeh, and L. Cranor. Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs. *Personal and Ubiquitous Computing*, pages 1–16, 2010.

[5] N. Bilton. Price of facebook privacy? start clicking. New York Times Article, May 12 2010.

[6] A. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 95–104, New York, NY, USA, 2010. ACM.

[7] P. Costa, J. Almeida, L. Pires, and M. van Sinderen. Evaluation of a rule-based approach for context-aware services. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1 –5, 30 2008-dec. 4 2008.

[8] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applicati. In *Human-Computer Interaction*, 2002.

[9] C. Hesselman, H. Eertink, and M. Wibbels. Privacy-aware context discovery for next generation mobile services. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, page 3, jan. 2007.

[10] G. Iachello, I. Smith, S. Consolovo, G. Abowd, J. Hughes, J. Howard, F. Potter, J. Scott, T. Sohn, J. Hightower, and A. LaMarca. Control, deception, and communication: Evaluating the deployment of a location-enhanced messaging service. In *Ubicomp '05*, pages 213 – 231. Springer-Verlag, 2005.

[11] A. Khalil and K. Connelly. Context-aware telephony: Privacy preferences and sharing patterns. In *CSCW '06*, 2006.

[12] P. Kumaraguru and L. F. Cranor. Privacy indexes: A survey of westin's studies. Tech report, Institute for Software Research International, Carnegie Mellon University, 2005.

[13] H. Nissenbaum. Privacy as contextual integrity. *Washington Law Review Association*, 79:119–158, 2004.

[14] L. Palen and P. Dourish. Unpacking "privacy" for a networked world. In *CHI '03*, pages 129–136, New York, NY, USA, 2003. ACM.

[15] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao. Understanding and capturing people's privacy policies in a mobile social networking application. *Personal and Ubiquitous Computing*, 13(16):401 – 412, August 2009.

[16] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh. Empirical models of privacy in location sharing. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 129–138, New York, NY, USA, 2010. ACM.

[17] V. Tuttlies, G. Schiele, and C. Becker. End-user configuration for pervasive computing environments. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on*, pages 487 –493, march 2009.