Work in Progress session at PerCom 2013, San Diego (19 March 2013)

Who, How, and Why? Enhancing Privacy Awareness in Ubiquitous Computing

Bastian Könings, Florian Schaub, and Michael Weber
Institute of Media Informatics
Ulm University, Germany
{ bastian.koenings | florian.schaub | michael.weber }@uni-ulm.de

***Abstract*—The combination and integration of sensing and interaction capabilities with almost ubiquitous inter-connectivity are basic requirements for context-aware systems to unobtrusively and invisibly support users in their daily activities. However, the invisible nature of such systems also threatens users' privacy. Users often lack awareness about a system's capabilities to gather data or to intervene in user activities, or even the system's presence. We propose a model to enhance user-centric privacy awareness by consistently modeling observations and disturbances of users. The model allows to capture *who* is affecting a user's privacy, *how* privacy is affected, and *why* it is affected. We further discuss how this model can be instantiated with discovery of channel policies.**

I. Introduction

Ubiquitous Computing (UbiComp) systems aim to support users in their activities by processing and interpreting context information in order to adapt to the user's needs [1]. UbiComp systems are typically characterized by being embedded, ubiquitous inter-connectivity, the ability to gather large amounts of sensor data, and their capability to autonomously act in the user's environment. Besides the benefits, these characteristics also carry non-negligible privacy implications for users [2]. Informational privacy as well as physical privacy [3] are harder to maintain in such UbiComp environments. Physical boundaries, such as walls, are not sufficient anymore to ensure privacy of a user's activity, because ubiquitous communication capabilities enable remote entities to access realtime information about the user's content and activity from anywhere. Physical and virtual environments start to merge, which impairs a user's privacy awareness, i.e., the ability to accurately perceive potential privacy threats. Especially threats stemming from virtual entities, which are not physically present, cannot be perceived by users without support. Thus, users may not be aware that they are being observed due to a mismatch between perceived physical boundaries and existing, but invisible virtual extensions of their environment. The focus of UbiComp systems on leveraging context information and implicit interaction to support user activities further impacts privacy, because the user's physical activities, state, and behavior need to be monitored continuously. Furthermore, UbiComp systems cannot only *observe* users, but can also physically *disturb* users and their activities, e.g., by autonomous interventions or audiovisual signals.

These privacy issues raise the need for a user-centric approach to improve and support privacy awareness and control in UbiComp systems. Existing approaches are mainly information-centric and neglect the outlined physical aspects. They often focus primarily on controlling privacy rather than also enhancing privacy awareness even though this problem was identified early on [4]. We only give a few examples. Hong and Landay [5] propose *Infospaces* to enable users to control the flow of information via in- and out-filters. Physical disturbances are not addressed and privacy awareness is limited to simple feedback when an entity explicitly requests access to some information. However, privacy awareness is an essential prerequisite for privacy decision making [6], [7], as it supports users in forming more accurate mental models of the system. Thus, it is necessary to find a trade-off between the invisibility of a UbiComp system and support of privacy awareness. Winkler and Rinner [8] propose mechanisms to achieve four levels of privacy awareness for surveillance cameras. In the highest level, users get attested feedback on the camera's deployed privacy mechanisms, e.g., image obfuscation. Here, privacy awareness is limited to physically present cameras and does not consider how video is forwarded to remote entities. Langheinrich [9] proposes a privacy awareness approach for UbiComp based on the concept of *privacy beacons* and P3P-based policies [10]. However, his approach does not consider physical disturbances and only partly addresses the virtual extension of physical boundaries. Entities in the user's proximity declare policies that allow to determine if an entity is forwarding information to others, but not how it is forwarded or handled by the other entity.

In this paper, we propose a user-centric model to support activity-based privacy awareness in UbiComp systems. Our model, outlined in Section II, reflects comprehensive virtual extensions of the physical environment by considering physically present and remote entities, as well as their observation and disturbance capabilities. The graph-based model allows to infer complex dependencies and interrelations between privacy affecting actions of entities and to reason about consequences of privacy control decisions. In order to instantiate the model we propose a policy discovery process in Section III. Section IV concludes the paper with an outline of future work.

II. A Model for User-centric Privacy Awareness

In order to support privacy awareness in UbiComp systems, solutions are required to discover the user's privacy state in the current situation and activity, and make relevant aspects

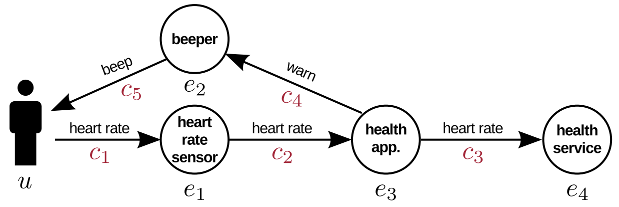

Fig. 1. Territorial privacy graph of a health monitoring use case.

of this state available through adequate user interfaces. Thus, we propose a model that can be instantiated by system level mechanisms and interpreted by visualization and interaction components on the user level.

In previous work [11], [12], we developed a territorial privacy model as a directed graph, which represents *who* potentially affects a user's privacy. A node in the graph represents an *entity*, which can be a person, but also any device, sensor, actuator, software agent, or service in the user's physical and virtually extended environment. Entities are either *active* or *passive*. Active entities either directly observe a user by gathering data about the user's context (e.g., a heart rate sensor) or disturb a user by generating noise or visual disturbances (e.g., an ambient speaker). Passive entities are connected to active entities through communication channels, represented as directed edges in the graph and called *observation channels* or *disturbance channels*. Consequently, active observing entities and all passive entities connected via observation channels are called *observers*; active disturbing entities and those connected via disturbance channels are called *disturbers*. An observation channel forwards gathered sensor data from one observer to another. Disturbance channels represent the flow of control commands and other events, which could lead to a physical disturbance, between disturbers.

As an example, Figure 1 shows a territorial privacy graph for a health monitoring use case. The graph consists of the user node u, entities $e_1 \ldots e_4$, and channels $c_1 \ldots c_5$. The user's heart rate is monitored by the health application e_3 (e.g., on the user's smartphone) via observation channel c_2 from the heart rate sensor e_1. In case of a critical heart rate, the user is warned by an acoustic signal. Thus, the health app triggers the beeper e_2 with disturbance channel c_4. Critical heart rates are also forwarded to the health service e_4 for the purpose of health statistics. This example shows how the territorial privacy graph can be used to model scenarios but also highlights some limitations. All entities that participate physically or virtually in a user's activity are represented; the channel content indicates *how* they could affect a user's privacy by observations or disturbances. However, the purpose of observations and disturbances (e.g, "health monitoring" or "critical heart rate warning"), forwarding conditions (e.g., forward heart rate only if critical) and channel dependencies (e.g., the disturbance channels depend on the observation channels) are not represented. How observed information is handled beyond the activity's context also remains unclear (e.g., are observations stored and used for marketing?). Yet, these aspects are important for comprehensive support of privacy awareness. Therefore, we extend the territorial privacy model, in the following, with the declaration of an entity's *purpose* for *actions* associated with the entity's channels. Actions allow to model explicit channel dependencies, as well as *conditions* and *obligations*, similar to information-centric access control concepts, e.g., in EPAL [13] or XACML [14].

A. Purpose

Purpose is a legal concept which binds data collection and usage to a predefined application context [2]. More generally, the purpose specifies *why* an entity affects a user's privacy with specific actions regarding observations and disturbances. In our model, the purpose consists of the *purpose domain*, the *purpose target*, and a *purpose description*.

Similar to P3P's primary-purpose [10], the domain describes the purpose's high level application category, e.g., *healthcare*, or *entertainment*. Domains can be hierarchically structured [15]–[17] to support abstraction levels and allow fine-grained specification of domains. For example, the healthcare purpose domain could have the subdomains *monitoring* or *vital assistance*. Thus, domain descriptions can be easily extended.

The target defines what the current purpose pertains to. For example, if an entity is supporting the user or her activity, the purpose target would be *user*. If an entity performs actions that support other entities, e.g., for marketing purposes, the corresponding entity would be declared as the purpose target. Thus, the purpose target allows to infer whether an entity is supporting the user or her activity, or some other entities. A purpose is also associated with a detailed purpose description that aids interpretation of the declared purpose.

B. Actions, Conditions, and Obligations

Actions model *how* an entity handles observation and disturbance channels. We identified a minimal set of seven actions deemed sufficient to represent different use and interrelations of channels. Observation channel related actions are *sense*, *forward*, *use*, and *store*. Disturbance channel related actions are *act*, *trigger*, and *control*.

The *sense* action always indicates the source of an observation. It refers to incoming observation channels of active observers (e.g., the physical channel between a user and a heart rate sensor). The *forward* action indicates that an entity's incoming observation channel is forwarded to one or more other entities, which potentially could access the channel's content (e.g., the sensed heart rate is forwarded to a health application). The *use* action indicates that an entity accesses the content of an incoming observation channel (e.g., the heart rate is analyzed). Similarly, the *store* action indicates that an entity stores a channel's content.

An *act* action always indicates the endpoint of a disturbance. It refers to outgoing physical disturbance channels of active disturbers (e.g., an acoustic signal of an ambient speaker). The source of a disturbance is either indicated by a *control* or a *trigger* action. A *control* action is specified by an entity for an outgoing disturbance channel when it is able to directly control an active disturber or an entity which in turn triggers an active disturber. For instance, a home automation system controlling the blinds of the user's room. A *trigger* action is declared by an entity for an outgoing disturbance channel when it creates events that cause disturbances (e.g., an application which triggers a warning). An active disturber can either be directly triggered or via multiple intermediate entities. For intermediate entities the *trigger* action has similar semantics for disturbances as the *forward* action for observations. The difference between *trigger* and *control* actions lies in their ability to control the type of disturbance. While a *trigger* action represents only an event without determining the type of disturbance (e.g., whether a warning is announced by an acoustic or visual signal), a *control* action is typically associated with a specific disturbance caused by the controlled entity (e.g., controlling the blinds of a room typically causes an audiovisual disturbance).

Each action can optionally have associated *conditions* and *obligations*. A condition specifies the required context dependencies in order to perform this action, e.g., a heart rate observation channel is only forwarded to the health service if the heart rate is critical. Thus, conditions provide a mechanism for context based privacy decisions. An obligation specifies what will be performed by an entity before, during, and after a given action. For instance, an obligation could state that an observation channel's content is modified in order to anonymize it before forwarding. Furthermore, obligations are used to declare an entity's treatment of observation channels after a user's activity has finished. For example, if an entity stores the content of an observation channel, an obligation could declare how long the content will be stored or if it is stored encrypted. Obligations and conditions must be adaptable by users for actions that relate to channels with user owned content, i.e., content gathered by devices or in environments possessed by the user (e.g., by the user's smartphone or in the user's home).

III. Model Instantiation

The proposed model allows to infer complex dependencies between entities and their actions. To instantiate the model in a practical system, we propose a discovery process based on *channel policies*, which describe the involved channels and their dependencies.

A. Channel Policies

A channel policy describes the set of incoming and outgoing channels of a particular entity, together with one or more purpose statements for different actions. A statement S consists of a purpose and a list of associated actions. A purpose is denoted as `<DOMAIN, TARGET, DESCRIPTION>`; an action is defined as `<ACTION {CHANNEL_SET}, {DEPENDENCIES}, CONDITION, OBLIGATION>`.

Figure 2 provides the channel policies for the health monitoring use case (see Fig. 1). The statement S_1 of entity e_1 involves actions *sense* for channel c_1 and *forward* for the channel set $\{c_1, c_2\}$, i.e., channel c_1 is forwarded to entity e_3 via channel c_2. The health application e_3 in turn specifies three statements S_1, S_2 and S_3. Statements S_1 and S_2 declare a purpose in the *healthcare* domain with subdomain *monitoring* and specify *user* as the target, which indicates that the actions of these statements are supporting the user. The *use* action indicates that the content of channel c_2 (the heart rate) is processed by the health app for the purpose "monitor heart rate". The *trigger* action indicates that the health app initiates the "beep" disturbance with channel c_4 for the purpose "warn user". Here, the statement specifies a condition "critical heart rate", which means that the disturbance is only triggered when this condition is true. The action further depends on S_1 because it is required as the contextual input to evaluate the condition. This implies that the *trigger* action also depends on all channels involved in S_1, which is only c_2 in this case. Statement S_3 specifies the same condition and thus also depends on S_1 in order to forward channel c_2 to c_3. Note, that a dependency could be declared as a set of statements or a set of channels. An example for the latter is the *act* action of the beeper e_2, which declares channel c_4 as a dependency. This indicates that the beeper is not able to autonomously initiate the action, and thus depends on the health app to do so. The health service declares an obligation for the *store* action, which states that the content of channel c_3 is stored encrypted and will be deleted after one year.

a) channel policy of entity e_1 (heart rate sensor)

channelSet(e_1) $= \{c^{hr}_{1\downarrow o}, c^{hr}_{2\uparrow o}\}$

$S_1(e_1) =$ <INFERRED>
<sense$\{c^{hr}_{1\downarrow o}\}, \{\}, -, -$>
<forward$\{c^{hr}_{1\downarrow o}, c^{hr}_{2\uparrow o}\}, \{\}, -, -$>

b) channel policy of entity e_2 (beeper)

channelSet(e_2) $= \{c^{warn}_{4\downarrow d}, c^{beep}_{5\uparrow d}\}$

$S_1(e_2) =$ <INFERRED>
<act$\{c^{beep}_{5\uparrow d}\}, \{c^{warn}_{4\downarrow d}\}, -, -$>

c) channel policy of entity e_3 (health application)

channelSet(e_3) $= \{c^{hr}_{2\downarrow o}, c^{hr}_{3\uparrow o}, c^{warn}_{4\uparrow d}\}$

$S_1(e_3) =$ <healthcare.monitoring, user, *"monitor h.r."*>
<use$\{c^{hr}_{2\downarrow o}\}, \{\}, -, -$>
$S_2(e_3) =$ <healthcare.monitoring, user, *"warn user"*>
<trigger$\{c^{warn}_{4\uparrow d}\}, \{S_1\}$,"critical heart rate", –>
$S_3(e_3) =$ <INFERRED>
<forward$\{c^{hr}_{2\downarrow o}, c^{hr}_{3\uparrow o}\}, \{S_1\}$,"critical heart rate", –>

d) channel policy of entity e_4 (health service)

channelSet(e_4) $= \{c^{hr}_{3\downarrow o}\}$

$S_1(e_4) =$ <healthcare.monitoring, user, *"provide critical h.r. stats"*>
<use$\{c^{hr}_{3\downarrow o}\}, \{\}, -, -$>
<store$\{c^{hr}_{3\downarrow o}\}, \{\}, -$,"encrypted, delete after 1 year">

Fig. 2. Channel policies for all incoming (↓) and outgoing (↑) observation (*o*) and disturbance (*d*) channels of the health monitoring use case.

The declaration of channel policies allows to infer existing dependencies and to reason about consequences of denying particular entities, channels, or actions. For example, if a user denies channel c_1 all dependent channels c_2, c_3, c_4 and c_5

together with associated purposes will get denied. It is further possible to infer missing purposes and combine purposes of dependent actions from a channel's source to its endpoint. For instance, the purposes "monitor heart rate", "warn user", and "provide critical heart rate stats" can be inferred for the heart rate sensor e_1 by following the observation channels c_1 and c_2, and interpreting the condition of $S_2(e_3)$ which shows that the disturbance channel c_4 depends on the observation channel c_2. Similarly, for the beeper e_2 the purpose "warn user" can be inferred from channel c_4.

B. Discovery Process

The goal of the discovery process is the collection of channel policies from all privacy affecting entities. We propose different discovery strategies depending on the user's environment. In a private environment like the home, we assume the deployment of trusted entities and thus rely on an optimistic approach based on entity collaboration. It is assumed that entities provide their own channel policies, comparable to how websites specify P3P policies [10]. A proactively initiated discovery process requests channel policies of all entities in the user's physical environment. If discovered policies contain channel references to further entities, those are subsequently queried for their respective channel policies. This way the discovery process can also include entities beyond the user's physical proximity. We are currently implementing this collaborative discovery approach based on UPnP[1].

For public environments, we propose a pessimistic discovery process as a combination of a community-based [8] and a beacon-based [9] approach. The beacon-based approach relies on interpreting broadcasted beacon messages in the user's proximity, which contain channel policies of entities in the physical environment. The advantage of this approach is that users can passively listen for those messages and do not need to trigger any discovery process. To address uncooperative entities, this approach can be further combined with a community-driven channel policy database. The database can be updated with user-provided channel policies about entities similar to [8] and either be queried for policies of specific entities or for policies at specific locations, e.g., all channel policies of observers at the user's current location. So far, we have implemented a privacy beaconing approach for broadcasting channel policies based on customized WiFi beacons. An Android based privacy client interprets received beacons and performs the subsequent discovery process which either queries further policies directly from referenced entities (if entities are collaborating) or from the community's database.

IV. Discussion & Outlook

The proposed user-centric model for privacy awareness considers privacy affecting observations and disturbances from entities in the user's proximity, as well as remote and virtual entities. The inclusion and discovery of this virtual extension is important to obtain a comprehensive view of the user's current privacy state in a specific situation and activity. Existing work on privacy awareness [8], [9] often considers only entities in the user's physical environment and neglects potential privacy disturbances [5]. Our model represents a user's privacy state at the system level and allows to infer complex dependencies and interrelations between privacy affecting actions of entities. The associated purposes of actions further allow to reason about consequences of privacy control decisions. We are currently investigating different optimistic and pessimistic strategies for policy discovery processes in order to instantiate the model, and plan to evaluate these approaches for different use cases in private and public environments. The model will further guide our next steps in developing user interfaces for privacy awareness that utilize the information provided by our model. We plan to integrate different abstraction levels which could visualize a user's privacy state depending on preferred information for privacy decisions, e.g., the type of entities, channel content, or purposes for different actions. The overall goal is to provide users with an intuitive yet comprehensive view of their current privacy state in order to support privacy decisions. The enforcement of these privacy decisions with adequate control mechanisms provides a further direction for future work. However, we argue that even without control capabilities, supporting privacy awareness of users is an important step towards privacy-friendly UbiComp systems.

References

[1] M. Weiser, "Some computer science issues in ubiquitous computing," *Communications of the ACM*, vol. 36, no. 7, pp. 75–84, 1993.

[2] M. Langheinrich, "Privacy by design - principles of privacy-aware ubiquitous systems," in *UbiComp '01*. Springer, 2001.

[3] S. D. Warren and L. D. Brandeis, "Right to privacy," *Harvard Law Review*, vol. 4, pp. 193–220, 1890.

[4] V. Bellotti and A. Sellen, "Design for privacy in ubiquitous computing environments," in *ECSCW '93*. Kluwer Academic Publishers, 1993.

[5] J. I. Hong and J. A. Landay, "An architecture for privacy-sensitive ubiquitous computing," in *MobiSys'04*. ACM, 2004, pp. 177–189.

[6] I. Altman, *The Environment and Social Behavior: Privacy, Personal Space, Territory, and Crowding.* Monterey, CA: Brooks/Cole, 1975.

[7] L. Palen and P. Dourish, "Unpacking "privacy" for a networked world," in *CHI '03*. ACM, 2003, pp. 129–136.

[8] T. Winkler and B. Rinner, "User-centric privacy awareness in video surveillance," *Multimedia Systems*, vol. 18, no. 2, pp. 99–121, Jul. 2011.

[9] M. Langheinrich, "A privacy awareness system for ubiquitous computing environments," in *UbiComp '02*. Springer, 2002, pp. 237–245.

[10] L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, M. Schunter, D. Stampley, and R. Wenning, *The Platform for Privacy Preferences 1.1*, W3C Std. P3P1.1, 2006.

[11] B. Könings, F. Schaub, M. Weber, and F. Kargl, "Towards Territorial Privacy in Smart Environments," in *Intelligent Information Privacy Management Symposium*. AAAI, 2010.

[12] B. Könings and F. Schaub, "Territorial Privacy in Ubiquitous Computing," in *WONS'11*. IEEE, 2011, pp. 104–108.

[13] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, *Enterprise Privacy Authorization Language (EPAL 1.2)*, W3C Std., 2003.

[14] E. Rissanen, "eXtensible access control markup language (XACML) version 3.0," OASIS, Standard xacml-3.0-core-spec-cs-01-en, 2010.

[15] G. Karjoth, M. Schunter, and M. Waidner, "Platform for enterprise privacy practices: Privacy-enabled management of customer data," in *PETS '03*. Springer, 2003, pp. 194–198.

[16] J. Byun, E. Bertino, and N. Li, "Purpose based access control of complex data for privacy protection," in *SACMAT '05*. ACM, 2005.

[17] Q. Ni, E. Bertino, J. Lobo, and S. B. Calo, "Privacy-Aware Role-Based access control," *IEEE Security & Privacy*, vol. 7, no. 4, pp. 35–43, 2009.

[1] http://upnp.org/sdcps-and-certification/standards