Distilling Privacy Requirements for Mobile Applications

Keerthi Thomas[1], Arosha K. Bandara[1], Blaine A. Price[1], Bashar Nuseibeh[1, 2]

[1] Centre for Research in Computing, The Open University, Milton Keynes, UK

[2] Lero, University of Limerick, Ireland

{keerthi.thomas, arosha.bandara, blaine.price, bashar.nuseibeh}@open.ac.uk

ABSTRACT

As mobile computing applications have become commonplace, it is increasingly important for them to address end-users' privacy requirements. Privacy requirements depend on a number of contextual socio-cultural factors to which mobility adds another level of contextual variation. However, traditional requirements elicitation methods do not sufficiently account for contextual factors and therefore cannot be used effectively to represent and analyse the privacy requirements of mobile end users. On the other hand, methods that do investigate contextual factors tend to produce data that does not lend itself to the process of requirements extraction. To address this problem we have developed a Privacy Requirements Distillation approach that employs a problem analysis framework to extract and refine privacy requirements for mobile applications from raw data gathered through empirical studies involving end users. Our approach introduces privacy facets that capture patterns of privacy concerns which are matched against the raw data. We demonstrate and evaluate our approach using qualitative data from an empirical study of a mobile social networking application.

Categories and Subject Descriptors

D.2.1 [**Requirements/Specifications**]: *Methodologies*; H1.2 [**User/Machine Systems**]: *Human factors*.

General Terms - Design, Security, Human Factors.

Keywords - privacy; mobile; requirements engineering

1. INTRODUCTION

The age of ubiquitous computing, particularly the rapid increase in the use of smart phones, has created a mass market for software applications that are being used in every context of users' daily lives. Previous research [1] has highlighted how system designers, policy makers, and organisations can easily become isolated from end-users' perceptions of privacy in different contexts. For mobile applications, end-users' context changes frequently and unpredictably, and observations of such users [33] suggest that changes in context result in changes in users' privacy requirements. Omitting these privacy requirements can affect users' privacy and consequently may have an impact on how well a system is adopted or utilised.

While knowledge acquisition techniques such as the use of Personas [5] have proven successful at dealing with the challenges of gathering and analysing the requirements of a large user base, the highly dynamic, and hard to predict usage scenarios associated with mobile applications still pose a challenge for existing requirements engineering approaches. This is particularly true for privacy requirements, which are known to be highly context-dependent [37] and are only likely to arise as users gain experience with an application [36]. This makes eliciting end-user privacy requirements for mobile applications both sensitive and difficult. Questionnaires do not elicit rich enough information about users' decisions and how these are influenced by the emerging context in a particular situation. To overcome such limitations, Goguen and Linde [24] proposed the use of ethnographic analysis techniques, such as conversation, discourse and interaction analyses to obtain tacit knowledge of what users actually do in different work situations. They also showed how the discourse analysis of users' stories can be used to explore the value systems of organisations and how the discourse analysis of users' explanations can be used for situated task analysis [23]. While Rubenstein [40] and Beyer and Holzblatt [9] have shown that shadowing of users is useful for capturing contextual requirements to design and build new systems, when it comes to privacy this direct approach is problematic, since the experience of being under constant observation is likely to change the behaviour of the users in ways that invalidate any observed behaviours with respect to privacy.

This prior work suggests that for mobile applications, privacy requirements are emergent requirements that need to be elicited and analysed from qualitative reports of the users' experience of the application. While there have been ethnographic studies conducted by the HCI community to study end-user privacy [36], [3], [7], including our own user studies [33], [34], user experience data from such studies does not readily translate into requirements. Often, this qualitative data, in the form of interview transcripts or user written reports, may contain privacy requirements that are embedded and tightly entwined with user's contextual experiences. The technical challenge in extracting these requirements systematically from the qualitative data relate to: (a) structuring and separating privacy relevant information from the qualitative data (b) identifying and extracting mobile privacy requirements from this data, and (c) modelling and representing the extracted mobile privacy requirements. Since privacy is a broad topic, we focus on *personal privacy* which refers to how people manage their privacy with respect to other individuals, as opposed to large organisations [26].

This paper makes two contributions, first, a novel framework is proposed for structuring problem analysis called *privacy facets*. The framework supports the identification of privacy requirements from different contextual perspectives – namely those of actors, information, information flows and places. It also uncovers privacy determinants and threats that a system must take into account in order to support the end-user's privacy. The second contribution is a technique called *requirements distillation* - a

ICSE'14, May 31 – June 7, 2014, Hyderabad, India

systematic method for qualitative data analysis that employs analysis models and patterns to extract and refine emergent software requirements, such as those relating to privacy for mobile applications. Distillation is a synthesis of thematic analysis from social sciences [10] and Problem Frames [27] from software engineering. Privacy requirements do not exist in a vacuum, rather they refer to other information requirements. Therefore, our approach to distilling privacy requirements makes the related information requirements manifest through the use of information problem patterns. Distillation makes use of both, information problem patterns and privacy facets to derive privacy requirements.

For the purpose of evaluation, we choose a pragmatic approach [17], because it is more aligned to our objective of engineering 'practical solutions' to real-world problems. Since distillation borrows techniques from both social sciences and software engineering, its evaluation is a mix of qualitative research methods [6][20] and case-study design [49]. Specifically, distillation is assessed for qualities such as: (a) employing a transparent and systematic process [49][22], (b) providing traceability by linking outputs to qualitative data [14][20], and (c) demonstrating applicability or usefulness of results [14] by informing system design.

In §2 we discuss some of the related research relevant to privacy requirements for mobile applications. §3 describes the overall process of requirements distillation and §4 presents our privacy facets (PriF) framework, which we use to enable a structured analysis of the privacy problems experienced by users. In §5 we describe a case study of mobile Facebook application and in §6 we discuss the results and limitations of the distillation process. §7 presents the available tool support. Finally, §8 concludes with ideas for future work.

2. RELATED WORK

Privacy management is a fine balancing act between what information is monitored, and the protections that are available against its search. As an enabler of both monitoring and searching, the architecture of mobile technology plays a key role in privacy [31]. In particular mobile application architectures incorporate numerous sensors (GPS, camera, accelerometer, etc.) that enable monitoring, together with ubiquitous network connectivity that enable continuous search and disclosure of monitored information. Further, the large screen displays of modern mobile devices facilitate proximal disclosures in public places. This leads us to view privacy as a constraint on the capabilities of the mobile application, and we adapt the concept of privacy described in Nissenbaum's Contextual Integrity framework [37] to define Mobile Privacy Requirements as: a set of constraints on a mobile computing application that enables appropriate flow of information depending on the user's context. Here the flow of information is the information sharing practices relevant to a user's context [44] and norms [28] that regulate it contribute to its appropriateness from end users' perspective.

While there has been significant work to understand privacy requirements based on laws and regulations (e.g., health-care information regulations – HIPAA [11]; and OECD guidelines [50], organisation privacy policy [13]), this research does not specifically address privacy violations experienced by mobile users. One of the ways to capture behaviour requirements for a software system is through the use of Use Cases. Seyff et al. developed a software environment called ART-SCENE, later extended to mobile applications [41], to discover and document stakeholder requirements by walking through scenarios that are automatically generated from use case specifications.

However, this approach is unlikely to work for studying mobile privacy because it is not practical to ask users to type their privacy requirements into a mobile device as they may be in transit or have limited input capabilities. Sutcliffe et al proposed a requirements elicitation framework (called PC-RE [45]) to describe functions that meet people's goals; characteristics of the users; and how users would like computer systems to achieve their personal goals. However, this work does not focus on privacy goals of end users.

The PriS method [29] uses eight categories of security and privacy principles to derive privacy requirements but these high-level principles are organisation-centric and do not cover fine-grained personal privacy threats end users face. In a similar approach, Deng et al. [15] have produced a threat taxonomy obtained by negating the main security properties. In their top-down LINDUUN approach, it may be difficult to a priori identify all potential privacy threats that are applicable to a software system. In Semantic Parameterization [12] privacy requirements were extracted from legal documents to produce a set of privacy requirements, however, these requirements are organisation-centric and do not specifically focus on personal privacy. Some propose the modelling of users' negative intent and behaviour as 'misuse cases' [4], others have used these to elicit security requirements [42] and privacy requirements for mobile applications [38]. Although, these approaches can potentially highlight deficiencies in a software system, it is difficult to anticipate all possible misuses of a mobile software system.

A number of researchers have investigated the uses of different types of qualitative data for requirements elicitation and design of mobile applications. For example, the user-centred Contextual Design method gathered a variety of data to develop a mobile application for baseball fans [25]. However, eliciting mobile privacy using this method will be problematic because shadowing of mobile end-users causes them to change their behaviour thus invalidating any requirements that were observed. In addition to this, privacy is a sensitive issue and often user's are not be able articulate their choices and decisions in an emerging context. A number of other studies [30], [8], [47], have used an enthnomethodological approach to elicit privacy requirements for mobile applications. Although these studies provided rich datasets that contained mobile privacy requirements, they did not provide mechanisms to structure and represent them such that they could be understood and implemented by software engineers and designers. The privacy requirements distillation technique described in this paper addresses this problem.

3. DISTILLATION PROCESS

As already discussed, mobile privacy has been studied by ethnomethodologists with the specific aim of producing new theories and high-level design guidelines. However, not all of these theories and guidelines have translated into concrete system requirements or design artefacts. The primary aim of distillation is to not only equip and assist software engineers with analytical tools and techniques but also provide process guidance on the extraction of privacy requirements from qualitative data which can be used in the design of privacy-aware software systems.

As a starting point, the distillation process relies on a software system that implements the initial requirements. This is the same software system for which qualitative data has been gathered and its user experiences are captured in the interview transcripts. The qualitative data and the initial systems requirements of the mobile application being studied form the two inputs to the distillation process (Figure 1).

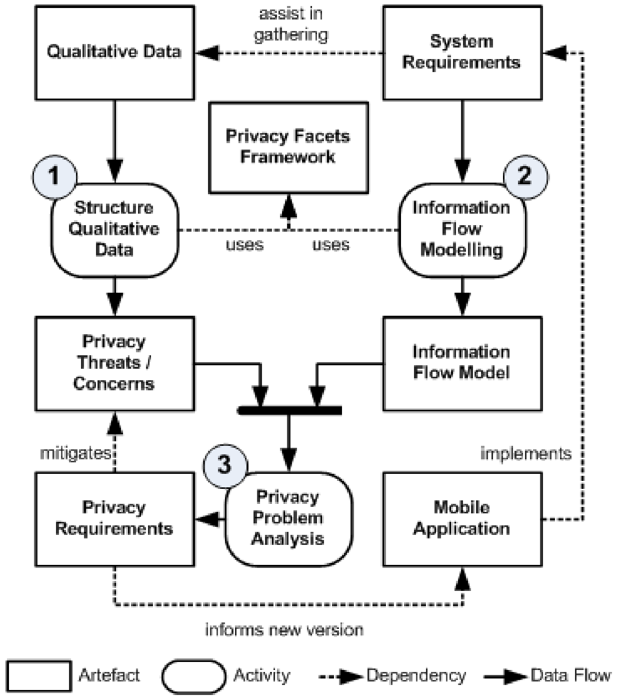


Figure 1. Privacy requirements distillation process

The distillation process consists of three main phases, namely: (1) Structuring of qualitative data; (2) Information-flow modelling; and (3) Privacy problem analysis.

Using an inductive approach inspired by Thematic analysis from social sciences, in the first phase, the qualitative data is structured using the Privacy Facets (PriF) framework, which provides pre-defined codes tailored for the identification of privacy-sensitive contexts. Once the privacy-sensitive contexts are isolated, additional codes from each facet of the PriF framework help in identifying the relevant privacy determinants and deriving the relevant privacy threats and concerns. The output of this phase is a set of privacy concerns experienced by users of the mobile application.

In the second phase, using initial systems requirements, problem models of information-flows are developed based on the information-flow problem patterns provided in the PriF framework. These problem models not only capture how the information is created but also how it is disseminated to other users.

The privacy problem analysis phase is the third phase where the privacy-sensitive context along with its privacy threats and concerns is analysed in conjunction with the information-flow problem models to identify the gaps in the current system, leading to the discovery of privacy requirements.

The distillation approach is designed to be a sequential process where the structuring of qualitative data precedes information-flow modelling and privacy problem analysis. In order to demonstrate our approach, we use data gathered from an empirical study of users of the mobile Facebook application[1], conducted in 2009 [33]. Although data from this study was analysed from an HCI perspective, it hadn't been previously analysed specifically for the purpose of requirements extraction as done in this paper. In this study, users were electronically shadowed in an unobtrusive manner and their responses to privacy issues were captured through an in-depth post-hoc interview. While the data we have analysed covers a range of functionality supported by the application, due to space limitations, in this paper concentrate on the participants' use of the 'Update status message' feature (Table 1).

Table 1. Example data from mobile Facebook study

[A.1.16] If I am out with friends I don't take my phone out, I don't do Facebook ...yes, ok, if I am with my sister I keep to read emails, but no I don't use Facebook and I tend not to use the mobile...because I am busy with other stuff, talking with them, socialising...Facebook tends to fill the gaps...if I am with a person I concentrate with that person.
[A.2.25] ...things like buses and trains I don't feel so comfortable..., because I don't know...lots of people I don't know...if they for example read some of the posts I have done...they don't know the people that they are aimed at or the back story...they'd probably come across quite differently and they would not understand them, it would look a little weird..[they would get] the wrong sort of almost the wrong first impression.
[A.3.42] anything I feel is private to myself I keep it to myself. I have a lot of good friends so if I want to share it I am happy to share it with all my friends. If there was something private, that is more close to me, like a girl that I liked and I wanted to share it with a friend I would do that in person rather than on Facebook

In the subsequent sections, we will demonstrate how the above data can be analysed using the requirements distillation method in order to derive privacy requirements for the mobile Facebook application.

4. PRIVACY FACETS FRAMEWORK

The main challenge of analysing qualitative data to derive privacy requirements is that the requirements will have to be systematically extracted from things that are not relevant but are tightly entwined with the users' experience, for example, the noise emanating from the operating context. To address this challenge, we propose a novel analytical framework, called *Privacy Facets (PriF)*, whose objectives are to provide: (a) analytical tools such as thematic codes, heuristics, facet questions and extraction rules to structure qualitative data; and (b) information-flow problem patterns and privacy arguments language to model privacy requirements.

4.1 Privacy-sensitive contexts

Thematic analysis is a method for identifying, analysing and reporting patterns or themes within qualitative data [10]. A 'theme' is said to capture something important about the data or having meaning within the dataset. When themes emerge within the data, they are encoded using appropriate *codes* (or labels) in a

[1] The mobile Facebook app has significantly changed since 2009.

process called 'coding' [14] or 'thematic coding' [20]. Similarly, distillation employs coding and makes use of specialised codes provided by the PriF framework to structure privacy related segments within the qualitative data.

Qualitative data may contain not only the users' experience but also their social interactions with other mobile users and actors in environment that may or may not be relevant to privacy. Therefore, the challenge of structuring this data would relate to isolating those aspects that are relevant to the extraction of privacy requirements. For this, the PriF framework provides a set of user-centric heuristics called *Negative Behaviour Patterns (NBPs)* and *Negative Emotional Indicators (NEIs)* to identify situations or settings involving privacy threats. We refer to the segments of qualitative data identified using these heuristics as a privacy-sensitive context or PS-context. This notion is adaption of 'context' from [16] where it is stated as 'any information that can be used to characterise the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves'. Therefore, PS-context refers to the location, identity and state of people, groups and computational and physical objects that affect end-users' privacy.

- NBPs are used to identify situations where users choose to not use (or ignore) an application due to privacy concerns (e.g., switching off all location services on their mobile device); or situations where the user completes a task that is supported by the application by some alternative means (e.g., communicating their location through a voice phone call rather than a location-based social network). This is based on the approach 'waving the red flag' and 'looking for the negative case' used by [14].
- NEIs are a set of key words that indicate the negative emotional state of the user in response to an event or action in the environment. For example, some of the key words are: concerned, unhappy, worried, scared, dislike etc. and include synonyms and semantically equivalent phrases. The presence of these NEIs in the qualitative data can indicate the presence of a privacy threat or concern. This is an adaptation of 'looking at emotions that are expressed and the situations that aroused them' used by [14].

Considering the example data from the mobile Facebook study (Table I), excerpt [A.1.16] would be coded as a NBP (based on the phrase *'I don't take my phone out, I don't do Facebook'*). Likewise excerpt [A.3.42] which includes an indication of a workaround (*'I would do that in person rather than on Facebook'*). Excerpt [A.2.25] on the other hand includes *'I don't feel so comfortable,'* indicating that is should be coded as a NEI.

4.2 Facet questions and privacy determinants

After extracting a PS-context from the data, the social aspects of the user's interaction have to be understood, for example, the actors involved, their roles and relationships with the user and the type of interactions that take place between them. To this end, the PriF framework proposes the use of 'facets' - a notion very similar to that of viewpoints [19] where each facet is considered to hold partial domain knowledge of the system. Since the knowledge is very specific to privacy, the facets are called privacy facets, each having unique properties and functions that must be analysed and addressed separately while at the same time be considered together to ensure completeness and consistency. There are four privacy facets namely: *Information*, *Information Flow*, *Actor* and *Place*.

Each facet can be used to gather specific domain knowledge that affects the privacy of mobile application users. The information facet elicits knowledge regarding *what* information is created by the software system, while the actor facet focuses on *who* the information is transmitted to, the information flow facet identifies *why* the information was transmitted and the place facet captures *where* the information was created or transmitted.

In the remainder of this section we describe the questions, privacy determinants and threats associated with each facet. For each privacy determinant, we also indicate the code used to annotate the qualitative data (e.g., [CODE(ATTR)]).

Information facet: Software systems produce data either by themselves (e.g. log transactions) or when the users interact with their functionality (e.g. take a digital photo). In order to be clear about how we relate data to information, we adopt Tenopir's definition of data and information in [52] which states: data are facts that are the result of observation or measurement and information is meaningful data or data arranged or interpreted in a way to provide meaning. When considering the privacy of information, we identify four questions that can be used to elicit the key privacy determinants relevant to this facet –

- Is the information personal or sensitive? - Personal information relates to a living individual who can be identified from that information. Sensitive information refers to information pertaining to an individual that can be used to characterise them in some way (e.g. religion, ethnicity, sexual orientation, etc.) [45]. Code: [I-TYPE(PERSONAL | SENSITIVE)]
- Is the information collected automatically (by computer automation) or manually (input by end-users)? - These two modes of information creation impact the types of privacy threats that can be discovered in the software system. For example, if the software system sampled certain information at a high frequency it can cause a surveillance effect. Code: [I-MODE(AUTO | MANUAL)]
- What is the purpose of the information or its context of use? - Knowing for what purpose the information is being collected is important, as it will help in later checking if the purpose was fulfilled or if it was used in a way detrimental to a user's privacy. Code: [I-PURPOSE]
- What are the information attributes? - Quality attributes can influence how the information is used within the system and perceived by its users. For example, some quality attributes could relate to accuracy (precision of data), completeness (all required data fields are filled), freshness (data is not expired and has become irrelavent), timeliness (data received at expected time frame i.e. within accepted latency), etc. Code: [I-ATTR(ACCURATE | COMPLETE | FRESH | ONTIME)

Actors facet: The actors facet pertains to the roles that a user can play in a given context and their relationships with other users. In the context of a software system, the roles of actors has a significant impact on the information-flows, thus understanding the roles of the actors (sender, subject and receiver); their relationships and responsibilities are critical to protecting privacy. For example, in a 'Hospital' context, the readings of patient's body temperature may be required by physicians to treat a health problem. The roles of both the patient and the physician with their roles, relationship and responsibilities should be understood and clearly defined. Privacy violations occur when these are ambiguous [2]. We identify the following questions for eliciting the key privacy determinants relating to the actors facet –

- What are the role relationships between the information sender, receiver and subject? - A role is the abstract characterisation of the behaviour of an active entity (or agent) within some context [35]. A relationship refers the relations between agents and corresponds to the social aspect of a role [28]. Together they determine the level of trust, which influences the sharing of sensitive information. Code: [ROLE(RELATIONSHIP)]
- What are the responsibilities associated with each role? - Responsibility is when one agent is responsible to another agent for something, and that this something can be described as a possible mismatch or non-conformance relation between an actual state of affairs and a desired, expected or feasible state of affairs (adapted from [28], p.87-106)). Responsibilities can affect the power relationships between actors, which in turn influences the information flows in a given context. Code: [ROLE(RESPONSBILITY)]

Information-flow facet: In order to understand the privacy requirements, all possible flows of information between the interacting users must be examined. Each of these information flows are governed by what Nissenbaum [37] calls transmission principles - informally established terms and conditions that guide the flow of information between different actors. In other words, transmission principles are constraints placed on the flow of information and breaching these constraints leads to a privacy violation. The following questions can be used to help elicit the key privacy determinants associated with this facet –

- What goals and purposes hold for information about a subject, flowing between the sender and receiver? – These transmission principles determine the privacy expectations of the subject, for example, if the subject needs to consent before the information is sent. Code: [FLOW(SENDER-SUBJECT)]
- What goals and purposes hold for information flows between the sender and receiver? – These transmission principles determine the privacy expectations of the sender and receiver, for example the sender would expect only certain receivers and not others. Code: [FLOW(SENDER-RECEIVER)]
- Are there any 3rd-party recipients of the information? – This determines the flow of information to 3rd-parties who can misuse the information. Code: [FLOW(3RDPARTY)]

Place facet: The place refers to a unique geographic location with a material form, meaning and value [21]. When mobile users move through different places, they interact with the objects that are present. Lessig [31] points out that the architecture at a given place influences privacy, in other words, the way in which the physical objects such as human agents and technologies are arranged in a place can have a direct impact on users' privacy; this was also shown by user studies from Mancini et al. [33]. Places can have their own set of rules or norms regulating social behaviours and interactions within them. Users are subjected to these rules, which may protect the privacy of others. The questions to be used to elicit the privacy determinants associated with the place facet are as follows –

- What are the places associated with the subject, sender and receiver? – Used to identify the places that can be associated with a privacy-sensitive context for the different actors. Code: [PLACE(LOCATION)]
- What norms apply to a place? – Used to identify the expected behaviours associated with a given place. Deviations from the expected behaviour can result in privacy threats being realised. Code: [PLACE(NORM)]

By asking the above questions of the example data from the mobile Facebook study (Table 1), we can apply the relevant codes for identifying the privacy concerns experienced by the user. The resulting coding is shown in Table 2, where for each example statement we have used different formatting to highlight the elements of the text that identify a PS-context, as well as the applicable privacy facets.

For instance, in statement [A.3.42] a privacy related context associated with a negative behaviour pattern (NBP) is identified due to the user saying 'I would do that in person rather than on Facebook'. Additionally, data elements relating to the information facet are identified in those portions of statement containing the text '*private*' and '*like a girl that I liked*', indicating that there is sensitive information being described in this context.

Table 2. Structured data from Mobile Facebook study

[A.1.16] If I **am out with friends I don't take my phone out,** I don't do Facebook ...yes, ok, if **I am with my sister I keep to read emails, but no I don't use Facebook and I tend not to use the mobile**...because I am busy with other stuff, talking with them, socialising...Facebook tends to fill the gaps...if I am with a person I concentrate with that person. **[NBP, PLACE(NORM)]**
[A.2.25] ...things like **buses and trains** I don't feel so comfortable..., because I don't know...lots of people I don't know...if they for example read some of the posts I have done...they don't know the people that they are aimed at or the back story...they'd probably come across quite differently and they would not understand them, it would look a little weird..[they would get] the wrong sort of almost the wrong first impression. **[NEI, PLACE(LOCATION)]**
[A.3.42] anything I feel is *private* to myself I keep it to myself. I have a lot of good friends so if I want to share it I am happy to share it with *all my friends*. If there was *something private*, that is more close to me, *like a girl that I liked* and I wanted to share it with a *friend* I would do that in person rather than on Facebook. **[NBP, I-TYPE(SENSITIVE), ROLE(RELATIONSHIP), FLOW(SENDER-RECEIVER)]**

Key to coded text: Privacy-sensitive context | *Information facet* | *Actor facet* | Information Flow facet | **Place facet**

In the next section we demonstrate how these codes can be used in the formulation of extraction rules for retrieving the qualitative data associated with different privacy threats. The data associated with each threat can then be used to identify gaps between the current software system and the users' expectation, leading to the discovery of privacy requirements.

4.3 Privacy threats and concerns

Parameters that influence privacy within a facet are likely to contribute to privacy threats; these are privacy violations that are likely to happen. When privacy threats are analysed in conjunction with the existing software system, its failings can be captured as privacy concerns which will have to be addressed in in a future version of the system. The PriF framework lists the possible threats and concerns that can be identified from the qualitative data.

As mentioned earlier, we define mobile privacy requirements as: a set of constraints on a mobile computing application that enables appropriate flow of information depending on the user's context. This notion of privacy is particularly suited for mobile applications because it takes context into consideration. Using the definitions used by Nissenbaum's Contextual Integrity framework [37], information-flow is described as a user (sender) transmitting information (or information attributes) about a subject to a user (receiver), while complying with a specific set of transmission principles.

For simplicity, in this paper we consider examples where the sender and subject are the same individual, but this need not always be the case. In general, transmission principles refer to the goals and purposes that govern the flow of information, encompassing the means and ends for which the information is being transmitted.

Madsen et al. [32] discuss several types of information flows in a software system that are relevant to addressing privacy, but this work concentrates only on those that are critical to addressing privacy in mobile applications which support peer-to-peer user interactions (e.g., mobile social networking applications). The majority of these applications are designed to make use of an intermediate application service provider(s) to facilitate information sharing among its users. Figure 2 shows a generic architecture containing three information-flows: information is created and sent to a service provider (F1), stored information is requested and is sent to a receiver (F2) and information sent to unintended receivers by either the service provider or the receiver (F3). The unintended receivers can also refer to actors who are co-located and in close proximity to the sender or receiver and is able to access the information without making a request to the software system.

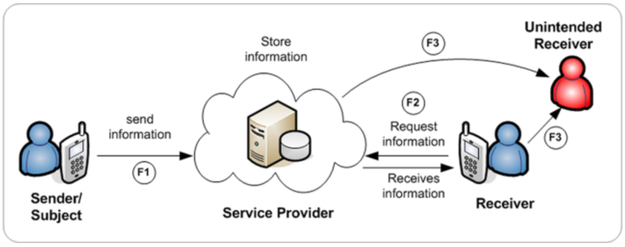


Figure 2. Information flows handled by the PriF Framework

A privacy violation is said to occur when an information-flow causes harm to the user, because of its faulty composition or because it is inappropriate. The components in an information-flow that influence privacy are called *privacy determinants* (e.g., sensitivity of information or role of the receiver). Privacy threats map information-flows in a software system to the harms a user

Table 3. Privacy threats, associated harms and data extraction rules

ID	Privacy Threat	Faulty information flows	Example Harms	Data Extraction Rule
T1	Identification	Subject's personal information is revealed.	Identity theft (H1) Financial loss (H2)	I-TYPE(PERSONAL) AND [NBP OR NEI]
T2	Exposure	Personal/sensitive information received by unintended recipients	Discrimination (H5) Loss of anonymity (H7) Relationship breakdown (H8) Embarrassment (H9) Physical danger (H10)	I-TYPE(SENSTIVE) AND [NBP OR NEI]
T3	Surveillance	Receiver makes frequent requests for information about the subject.	Emotional harm (H4) Loss of freedom (H6) Physical danger (H10)	I-MODE(AUTO) AND [NBP OR NEI]
T4	Aggregation	Receiver combines datasets to produce a new type of information without the subject's consent.	Discrimination (H5)	I-PURPOSE AND [NBP OR NEI]
T5	Misinformation	Inaccurate or insufficient level of information about the subject is transmitted.	Loss of reputation (H3) Emotional harm (H4) Discrimination (H5)	I-ATTR(*) AND [NBP OR NEI]
T6	Breach of trust	Receiver forwards the information to others contravening the subject's terms and conditions.	Loss of reputation (H3) Emotional harm (H4)	ROLE(RELATIONSHIP) AND [NBP OR NEI]
T7	Power imbalance	Receiver uses information to control the subject.	Loss of freedom (H6) Relationship breakdown (H8)	ROLE(RESPONSIBILITY) AND [NBP OR NEI]
T8	Cross-contextual information flow	Information from one context may be used in another context	Loss of reputation (H3) Discrimination (H5)	FLOW(*) AND [NBP OR NEI]
T9	Proximal access	Unintended receivers can access information due to close physical proximity to the sender or receiver.	Loss of reputation (H3) Loss of freedom (H6) Loss of anonymity (H7) Embarrassment (H9)	PLACE(LOCATION) AND [NBP OR NEI]
T10	Intrusion	Information flow disturbs receiver's tranquility.	Emotional harm (H4) Loss of freedom (H6)	PLACE(NORM) AND [NBP OR NEI]

can suffer and privacy threats when realised cause privacy violations. *Privacy concerns* describe the gap between the requirements model (or its implementation) and the identified privacy threats. Privacy requirements address these privacy concerns by providing suitable feedback and control facilities such that the user has better control over the information-flows, which are linked with specific privacy threats.

The privacy taxonomy proposed by Solove [43] has sixteen types of privacy violations that are broadly applicable to software systems, however they do not necessarily focus on privacy violations that are possible when using mobile applications. One of our previous empirical studies involving mobile users [33] identified privacy violations related to the use of mobile applications. Combining both these contributions, we refine and present the privacy threats applicable for mobile software systems, together with the potential harm associated with each threat (Table 3).

This taxonomy of privacy threats and harms that we have developed links the privacy threats that can arise due an inappropriate information flow to the potential harm that can result to the end user. For example, a mobile application that allows sensitive information to flow to an unintended recipient will create an exposure threat (T2) that might result in discrimination (H5), loss of anonymity (H7), relationship breakdown (H8), embarrassment (H9) or physical danger (H10) to the end user.

Table 3 also shows the data extraction rule associated with each threat, which can be used to retrieve the qualitative data that matches the combination of codes given in each rule.

For instance, executing the extraction rule for the exposure threat (T2) will return the excerpt [A.3.42] from Table 2, which is coded with both I-TYPE(SENSTIVE) and NBP. This indicates that the mobile Facebook application causes the user to report a negative behaviour when sensitive data is associated with a data flow. This privacy threat arises because the application is unable to detect the information type or limit its flow to a subset of the user's friends.

4.4 Information-flow problem patterns

Since privacy requirements are related to information-flows in a software system these must be modeled as part of the requirements distillation process. The PriF framework uses information-flow problem patterns for this purpose. The first part of the information-flow relates to how information is created. The PriF framework captures this aspect as an *information creation* problem pattern while the second aspect is captured as an *information dissemination* problem pattern, both of which are based on the Problem Frames method [27]. We have chosen Problem Frames for our analytical framework because it supports a notion of context where real world domains (i.e. physical domains) are explicitly modelled which are critical to understanding privacy in mobile applications.

Information-flow in its simplest form consists of a sender, receiver and the information that is transmitted between them. As privacy relates to flow of personal/sensitive information, the subject of the information should also be considered. In addition, information-flows have goals and purposes to achieve, which play an important role in the flow of personal information from the sender to the receiver [37].

Putting all these together we define a privacy problem in an information system as that of: building a machine that will allow appropriate flow of personal information and/or avoid inappropriate flow of personal information (i.e. avoid privacy violations); where the appropriate or inappropriate flow of personal information is a function of the information type, roles of actors and transmission principle.

A composite model of the information-flow problem frame for the 'Status update' feature of the mobile Facebook application is shown in Figure 3. It is composed of two smaller sub-problems: information creation (IC) problem and information dissemination (ID) problem.

In the information creation problem, the mobile phone acts as connection domain between the user and the machine. In the problem frame diagram (Figure 3) the user is a biddable domain representing the human operator. To create the status message, the user issues a create command Create(SM) at interface a, which is executed by the machine, sending equivalent commands to the model domain status message where it is stored. The User issues commands Update(SM) and Delete(SM) respectively to perform further updates and deletions to the status messages.

In the second part of the problem, the emphasis is on how the information reaches the recipients, therefore modelling of information dissemination deals with the viewing or receiving of the information. Normally, users are able to view information when they make queries to the software system, however the mobile Facebook application is designed to display a user's status message to their friends as soon as the application is launched.

Therefore, the friends of the user are able to view the status message when they log into the software system because the system automatically makes a request, the command Request(SM) at interface g and the message answering machine responds to the query by reading the information from the model domain Status Message at interface e and updates the mobile display accordingly.

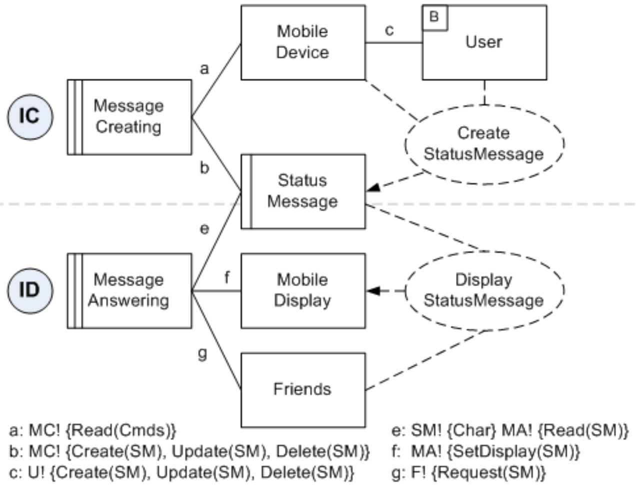


Figure 3. Information flow problem frame for 'status update'

With the basic information flows of the 'Update Status' feature modelled, we can now analyse the control variant of this information-flow model where the emphasis is on the controlling of information and the rules that govern its dissemination. Therefore, in the control variant problem frame (Figure 4) the central feature is a privacy rules model domain, which contains rules for information creation and editing; and also rules for answering queries for information.

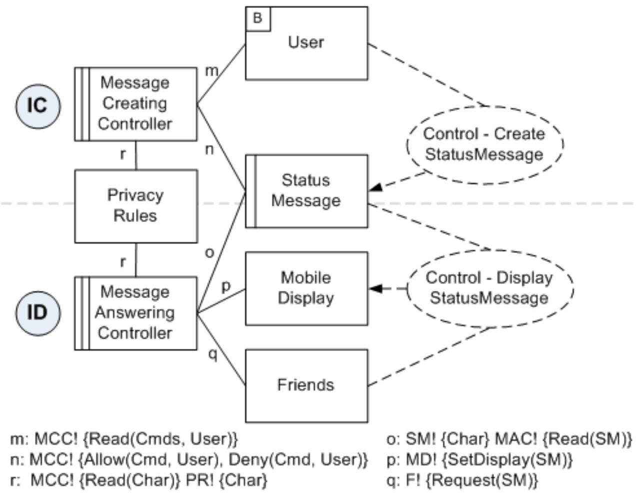


Figure 4. Control variant problem frame for 'status update'

The message creating controller checks the privacy rules to determine if the user is allowed to issue the commands to create, modify, delete or forward status messages. Similarly, when the friends make request for the status message, the message answering controller checks the privacy rules to determine if the request should be answered or not, setting the mobile screen display accordingly.

While the above problem frames modelled the information-flow within the software system, its control variants help in constraining how the information is created and disseminated, thus addressing privacy. Out of the five basic problem frames found in the Problem Frames method [27], the information creation problem is a modification of the basic workpiece problem frame while the information dissemination is fitted to an operator variant of an information problem frame. In the problem frames method, each problem has a frame concern, which highlights a certain aspect of the problem demanding the attention of the analyst/developer.

Similarly, in the PriF framework the privacy concern is simply a special type of frame concern relating to privacy, which the analyst must take into account and address in the software system to make it privacy-aware. Therefore, the privacy concerns extracted from the qualitative data through the use of facets should be addressed in order to support the privacy of end users.

4.5 Privacy Arguments

In previous work we developed a notation for mobile application privacy requirements called *privacy arguments*, that could: be used to reason about privacy requirements; be integrated into a mobile application to enable users to fine-tune the requirements at run-time; and provide run-time diagnostics about the satisfaction of privacy requirements [48]. However, there was no process to support the formulation of privacy argument classes. We have integrated privacy arguments into the PriF framework, extending it to link the requirements derived through the distillation process to the original qualitative data.

From a developer's perspective, a privacy argument justifies to an audience, such as users of mobile applications, that the user's privacy claim has been respected by the software system. The general structure of privacy arguments is: (Warrant, Ground → Claim). Privacy requirements described as the claim of an argument that needs to be justified. The ground is the collection of facts that can be observed from the world domains, which supports the claim. The warrant is the collection of domain-specific rules that links the ground to the claim of the argument. The developer can formulate privacy arguments as argument classes, which are instantiated by the user with specific parameter values that are appropriate to their particular context.

5. CASE STUDY: MOBILE FACEBOOK

In section 4.3 we explained how the qualitative data was used to highlight a privacy concern relating to the threat of sensitive data being exposed. This concern arose because the software system is unable to distinguish between sensitive and non-sensitive information, resulting in sensitive information being visible to all friends. Using the PS-context that was extracted to derive this concern [A.3.42] we can express this as shown in Listing 1.

Here the argument class MFb_CloseFriends_Norm captures the user's intended behaviour as described by the qualitative data, whereas the argument MFb_Exposure_Concern specifies the behaviour of the system that causes the user to exhibit a negative behaviour pattern. To address this concern, we could build a machine that would automatically check if the information is sensitive or not, otherwise the machine could prompt to the user to make the decision.

By being able to determine information sensitivity, the software system will be able to ensure that sensitive information is not visible to any unintended recipients (e.g., in this case, if the user selects a group of friends/recipients who should not be viewing the sensitive information, then the software system can immediately alert the user of the potential threat). In this way, we can mitigate the effect of exposure where sensitive information cannot be leaked to a wider unintended audience.

```
argument: MFb_CloseFriends_Norm
PN1 "<<User>> can only share Status Messages
with close friends" {
  supported by
    F1 "<<User>> has close friends"
    F2 "<<User>> creates sensitive msg"
    F3 "<<User>> wants sensitive msg to be
         seen by close friends only"
    F4 "Close friends want to see sensitive msg"
  warranted by
    R1 "<<User>> inputs sensitive msg"
    R2 "When a close-friend taps the Fb icon
        on his mobile device, the application
        opens with sensitive msg displayed”}

argument:  MFb_Exposure_Concern
PC2 “Status messages are considered as non-
     sensitive by the system” rebuts PN1 {
  supported by
    F6 “User is unable to classify a status
        message as being sensitive or non-
        sensitive”
    F7 “The system is unable to differentiate
        between sensitive and non-sensitive
        status message”}
```

Listing 1. Privacy norm and exposure concern argument

In order to mitigate the exposure concern, the requirement for checking of information sensitivity is captured in the following argument construct (Listing 2).

```
argument: MFb_Inf_Sensitivity_Detect
PR1 "Status message sensitivity can be
    detected" mitigates PC2{
  supported by F2
  warranted by
    Cr1 "System detects sensitivity of msg:"
        ? SensitiveMessage(StatusMsg)
    Fr2 "If Cr1 is indeterminate, ask user to
        select sensitivity label”}
```

Listing 2. Privacy argument to check information sensitivity

Detecting the creation of information sensitivity is just one part, the other part relates to information receivers. From the problem context [A.3.42], it is evident that the user wished to share sensitive status messages only with close friends but the software system did not facilitate the creation of such groups. Therefore, the next privacy requirement is about allowing the user to create a recipient group called ‘close friends’ (Listing 3).

```
argument: MFb_Close_Friends_Group
PR2 "<<User>> can create group of close
    friends" mitigates PC2{
  supported by F1
  warranted by
    Cr3 "<<User>> issues command:”
         CreateGroup(CloseFriends)”
    Fr2 "<<User>> assigns friends to group”:
         AssignGroupMember(f, CloseFriends) }
```

Listing 3. Privacy argument to create ‘close friend’ group

On its own, the privacy requirements PR1 and PR2 may not be sufficient to mitigate privacy concern PC2 because it does not take into consideration the recipients who will receive the sensitive information. Another requirement regarding query answering must be defined such that only those who are members of the close-friends group may be allowed to see status messages marked as being sensitive. This is done by an additional requirement in PR3 as shown in Listing 4.

```
argument: MFb_Close_Friends_Viewing
PR3 "Only close friends of <<User>> can see
     sensitive status messages"
  mitigates PC2{
  depends on PR1, PR2
  warranted by
    Cr5 "Senstive status msg only visible to
         close friends”:
         IF SensitiveMessage(m)&
            CloseFriends(cf,User) THEN
            StatusMessageView(m,cf)}
```

Listing 4. Privacy argument for information dissemination

We use the depends on clause to indicate that PR3 has a dependency on other requirements such as PR1 - the system’s ability to determine if the status message was sensitive or not and PR2 – the user’s ability to create a group (list) of close friends. As shown, a privacy requirement can mitigate one or more privacy concerns and similarly a privacy concern can rebut one or more information-flow norms in a software system. This third phase of distillation showed the derivation of privacy requirements for mobile applications, in the form of privacy arguments, for a single privacy concern associated with the threat of exposure (T2). A similar analysis process can be carried out to yield requirements relating to other concerns derived from the qualitative data.

6. DISCUSSION

Distillation not only follows a systematic approach but also its output in the form of privacy arguments can be traced back to its source in the qualitative data. Further, the information-flow problem models with their associated privacy arguments aimed to address the gaps in the software system studied and therefore protect the privacy of end-users. However, in this section we discuss some of the factors that influence the validity and limitations of our approach.

The first limitation of our approach is scalability. Distillation uses a number of different analysis techniques. For example in order to apply privacy requirements distillation, software engineers need to be familiar with qualitative data analysis techniques, which is not the norm. Whilst acquiring the necessary data coding skills is not difficult, it takes practice to get it right with the analysts’ level of expertise influencing the quality of output. This can be managed through the use of software tools and templates, which can reduce the complexity and improve the outcome of the analysis. One way to encourage the use of distillation would be to provide software tools to assist the analyst. In the next section we describe some of the tooling options we have explored in order to support the distillation process.

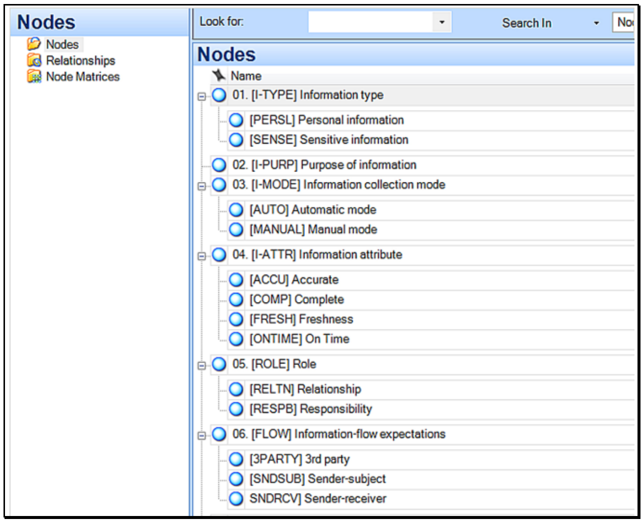
The second factor relates to reliability. Similar to other inductive approaches, thematic coding in the distillation approach is subjective and depends on the software engineer's interpretation of raw data. This implies that identification of PS-contexts, privacy threats and concerns can be biased. Inductive approaches prescribe the use of an assessment process where an initial coder produces a set of codes and additional analysts may be asked to apply these codes to the same raw data. The variations between the initial coder and subsequent ones are statistically measured to prove the reliability of codes [18][46], a similar assessment needs to be carried out on distillation, Although, it may not be difficult to train a group of software engineers to use our approach, to overcome any initial inter-coder disagreements, software engineers can be encouraged to discuss and agree with each other's interpretations of the raw data, similar to code cross-checking [20]. While we acknowledge such inter-coder assessments can improve the confidence and reliability of our approach, this is considered to be future work.

The third factor relates to distillation’s generalisability. The qualitative data from the mobile Facebook study had two dimensions, namely (a) mobility of users (b) personal privacy. Although distillation and more specifically the PriF framework had been designed to analyse these two dimensions, we believe the approach can cover scenarios where the dimension of mobility is not included in the input, for example, qualitative data from studies involving Facebook users with no reference to their mobility. In such cases, the place facet in the PriF framework may not be fully utilised. But for the approach to be successful the underlying privacy norms that produce negative behaviour patterns (NBPs) and emotions (NEIs) in users should be captured in the qualitative data. This leads us to conclude that distillation critically relies on NEIs and NPBs within the qualitative data to analyse privacy requirements and without these markers it will be difficult to apply this approach on other datasets. Therefore, distillation cannot be generalised for qualitative data that does not have a strong focus on privacy.

The last factor relates to completeness. The application of distillation demonstrated how the approach can help software engineers derive privacy requirements that address end-users' privacy concerns. While the derived requirements could be used to improve the design of privacy functionality of the software system that was studied, it was not possible to validate this by modifying the software and testing it with the users again. Using distillation in an iterative software development project, where the effectiveness of the derived requirements can be evaluated empirically remains an area for future work.

7. TOOL SUPPORT

There were two main requirements for automated tool support of our privacy requirements distillation process. First, it should support the extraction of privacy concerns from qualitative data, and second, it should support the modelling of information flows of the current software system and later help in the privacy problem analysis. Both of these activities are based on two well-known and proven methods: qualitative data analysis and the problem frames method respectively. As a first step towards supporting software engineers in distilling privacy requirements we decided to customise existing tools from each of these areas.



01. [I-TYPE] Information type
[PERSL] Personal information
[SENSE] Sensitive information
02. [I-PURP] Purpose of information
03. [I-MODE] Information collection mode
[AUTO] Automatic mode
[MANUAL] Manual mode
04. [I-ATTR] Information attribute
[ACCU] Accurate
[COMP] Complete
[FRESH] Freshness
[ONTIME] On Time
05. [ROLE] Role
[RELTN] Relationship
[RESPB] Responsibility
06. [FLOW] Information-flow expectations
[3PARTY] 3rd party
[SNDSUB] Sender-subject
SNDRCV] Sender-receiver

Figure 5. NVivo codes for PriF framework

From the qualitative data analysis domain, we chose to use Nvivo (http://bit.ly/NVivo) because of its prevalence and wide use in qualitative research. One of the main advantages of using NVivo is that the codes from the PriF framework can be pre-defined and stored to be re-used by other projects/analysts. The pre-defined codes in NVivo serve as a template for the structuring of qualitative data, making it easier for the analyst to readily apply the codes on the transcribed texts. In NVivo, each node represents a code or concept. Therefore, all the codes used in coding phase of distillation - NEIs, NBPs and privacy facets (Section IV) are defined as a hierarchy of nodes in Nvivo (Figure 5). The tool also supports the specification of extraction rules, thus automating identification of the privacy-related contexts associated with different threats.

For privacy problem analysis, we chose to use OpenArgue [51], an Eclipse plug-in that supports both problem frames modelling and incremental arguments written in propositional logic. The tool can perform syntax checking, visualizing, formalizing, and reasoning over these incremental arguments. OpenArgue integrates a 'decreasoner', which is an off-the-shelf reasoning tool that translates propositional formulae into problems for SAT-solvers. The integrated tool supports logical deduction to check whether an argument is valid, and model finding to obtain counterexamples to the argument. On the basis of these results, rebuttals and mitigations are generated and visualized.

While the use of the off-the-shelf tools was adequate to testing the concepts behind the privacy requirements distillation method, the lack of integration between the tools was a drawback. Additionally, to be useful in real-world mobile applications development projects it will be necessary to develop tools that integrate directly into standard development environments such as Eclipse. Since the source code for OpenArgue is freely available, and it is already integrated into Eclipse, our strategy is to extend this tool to support the qualitative data analysis needs of the requirements distillation approach. Development of this tool remains an area of future work.

8. CONCLUSIONS

Eliciting mobile privacy requirements is challenging, largely due to the fact that mobile privacy issues are so dependent on the physical and socio-cultural context of the users. This means that only data that captures the nuances of these contextual factors and variations can adequately inform the development of privacy requirements for privacy-aware mobile applications. The distillation approach we proposed in this paper allows requirements analysts to take advantage of the richness of qualitative empirical data while refining this data systematically into a form that enables it to be used for the design of mobile applications that reflect users' real privacy concerns and needs. Our distillation process uses a novel privacy facets framework to structure raw data and to derive privacy concerns.

To support the privacy distillation we have adapted off-the-shelf tools such as NVivo and OpenArgue. Further automated tools can help the software engineer integrate different phases of the distillation process into standard software development environments such as Eclipse. In addition to undertaking work to address the limitations discussed above, we intend to conduct further evaluations of our approach by using other sources of empirical data, such as our studies of location tracking [34].

9. ACKNOWLEDGMENT

This research was partially funded by a Microsoft Software Engineering Innovation Foundation (SEIF) Award, Science Foundation Ireland grant 10/CE/I1855 and by the European Research Council (Advanced Grant 291652 – ASAP).

10. REFERENCES

[1] A. Adams and M. A. Sasse. Privacy issues in ubiquitous multimedia environments Wake sleeping dogs, or let them lie. In *Proceedings of INTERACT 99*, Edinburgh, 1999, pp. 214–221J.

[2] A. Adams. Users' perception of privacy in multimedia communication. In *CHI'99 extended abstracts on Human factors in computing systems*, Pittsburgh, Pennsylvania, 1999, pp. 53–54.

[3] A. Adams. Multimedia information changes the whole privacy ballgame. In *Proceedings of 10th conference on Computers, freedom and privacy: challenging the assumptions,* Toronto, Ontario, Canada, 2000, pp. 25–32.

[4] I. Alexander. Misuse cases: use cases with hostile intent. *Software, IEEE,* 20 (1). 58-66.

[5] M. Aoyama. Persona-and-scenario based requirements engineering for software embedded in digital consumer

products. In *Proceedings of 13th IEEE International Conference on Requirements Engineering,* 2005., pp. 85-94.

[6] J. Baxter and J. Eyles. Evaluating Qualitative Research in Social Geography: Establishing 'Rigour' in Interview Analysis. *Transactions of the Institute of British Geographers*, 22 (4). 505-525.

[7] V. Bellotti and A. Sellen. Design for privacy in ubiquitous computing environments. In *Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work*, Norwell, MA, USA, 1993, pp. 77–92.

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

[9] H. R. Beyer and K. Holtzblatt, Apprenticing with the customer. *Communications ACM*, vol. 38, no. 5, pp. 45–52, 1995.

[10] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3 (2). 77-101.

[11] T. D. Breaux and A. I. Anton. Analyzing Regulatory Rules for Privacy and Security Requirements. *IEEE Transactions on Software Engineering,* vol. 34, no. 1, pp. 5–20, 2008.

[12] T.D. Breaux and A.I. Anton. Mining rule semantics to understand legislative compliance. In *Proceedings of the workshop on Privacy in the electronic society*, (Alexandria, VA, USA, 2005), ACM, pp. 51 – 54

[13] T.D. Breaux and A. Rao. Formal analysis of privacy requirements specifications for multi-tier applications. In *21st IEEE International Requirements Engineering Conference (RE)*, 2013, pp. 14-23.

[14] J. Corbin and A. Strauss. Basics of Qualitative Research, Techniques and Procedures for Developing Grounded Theory, *3rd ed. Sage Publications*, 2008.

[15] M. Deng, K. Wuyts, R. Scandariato, B. Preneel and W. Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16 (1). 3-32.

[16] A.K. Dey. Understanding and Using Context. *Personal Ubiquitous Computing*, 5 (1). 4-7.

[17] S. Easterbrook, J. Singer, M-A. Storey and D. Damian. Selecting Empirical Methods for Software Engineering Research. In *Guide to Advanced Empirical Software Engineering*, Springer London, 2008, 285-311.

[18] J. Fereday and E. Muir-Cochrane. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International Journal of Qualitative Methods*, 5 (1). 80-92.

[19] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke. Viewpoints: A Framework for Integrating Multiple Perspectives in System Development. *International Journal of Software Engineering and Knowledge Engineering*, vol. 2, no. 1, pp. 31–58, 1992.

[20] G.R. Gibbs. Analyzing Qualitative Data. *SAGE Publications*, London, England, 2007.

[21] T. F. Gieryn. A space for place in sociology. *Annual Review of Sociology*, vol. 26, no. 1, pp. 463–396, 2000.

[22] N. Golafshani. Understanding reliability and validity in qualitative research. *The Qualitative Report*, 8 (4). 597-607.

[23] J. A. Goguen. The Dry and the Wet. In *Proceedings of the IFIP TC8/WG8.1 Working Conference on Information System Concepts: Improving the Understanding*, 1992, pp. 1–17.

[24] J. A. Goguen and C. Linde. Techniques for requirements elicitation. In *Requirements Engineering*, 1993., Proceedings of IEEE International Symposium on, 1993, pp. 152–164.

[25] K. Holtzblatt. Customer-centered design for mobile applications. *Personal Ubiquitous Computing*, vol. 9, no. 4, pp. 227–237, 2005.

[26] G. Iachello and J. Hong. End-user privacy in human-computer interaction. *Foundations and Trends in Human-Computer Interaction*, 1 (1). 1-137.

[27] M. Jackson, Problem frames: analyzing and structuring software development problems. *Addison-Wesley Longman Publishing Co., Inc.*, Boston, MA, USA, 2001.

[28] M. Jirotka and J. A. Goguen. Requirements engineering: social and technical issues. *Academic Press Professional, Inc.*, San Diego, CA, USA, 1994.

[29] C. Kalloniatis, E. Kavakli and S. Gritzalis. Using Privacy Process Patterns for Incorporating Privacy Requirements into the System Design Process. In *The Second International Conference on Availability, Reliability and Security*, 2007, 1009-1017.

[30] A. Khalil and K. Connelly. Context-aware telephony: privacy preferences and sharing patterns. In *Proceedings of the 20th anniversary conference on Computer supported cooperative work*, Banff, Alberta, Canada, 2006, pp. 469–478.

[31] L. Lessig. The Architecture of Privacy. *Vanderbilt Entertainment Law and Practice*, vol. 1, no. 56, pp. 63–65, 1999.

[32] P. Madsen, M. C. Mont, and R. Wilton. A Privacy Policy Framework – A position paper for *the W3C Workshop of Privacy Policy Negotiation*, vol. 2012. 2006.

[33] C. Mancini, K. Thomas, Y. Rogers, B. A. Price, L. Jedrzejczyk, A. K. Bandara, A. N. Joinson, and B. Nuseibeh. From spaces to places: emerging contexts in mobile privacy. In *Proceedings of the 11th international conference on Ubiquitous computing*, Orlando, Florida, USA, 2009, pp. 1–10.

[34] C. Mancini, Y. Rogers, K. Thomas, A. N. Joinson, B. A. Price, A. K. Bandara, L. Jedrzejczyk, and B. Nuseibeh. In the Best Families: Tracking and Relationships. In *Proceedings of the 29th International Conference on Human Factors in Computing Systems, ACM CHI 2011*, 2011.

[35] F. Massacci, J. Mylopoulos and N. Zannone. Security Requirements Engineering: The SI* Modeling Language and the Secure Tropos Methodology. In *Advances in Intelligent Information Systems*, Springer Berlin Heidelberg, 2010, 147-174.

[36] D. H. Nguyen, A. Kobsa, and G. R. Hayes. An empirical investigation of concerns of everyday tracking and recording technologies. In *Proceedings of 10th International*

conference on Ubiquitous computing, New York, USA, 2008, pp. 182–191.

[37] H. Nissenbaum, Privacy in Context: Technology, Policy and the Integrity of Social Life. *Standford University Press*, Standford, California, 2010.

[38] I. Omoronyia, M. Salehie, R. Ali, H. Kaiya and B. Nuseibeh. Misuse case techniques for mobile privacy. In *1st International Workshop on Mobile Privacy Management (PriMo2011)*, 2011, Copenhagen, Denmark.

[39] Parliament, British. 'Data Protection Act of 1998'. 1998.

[40] H.B. Reubenstein, and R.C. Waters. The Requirements Apprentice: automated assistance for requirements acquisition. In *IEEE Transactions on Software Engineering*, 17 (3). 226-240

[41] N. Seyff, F. Graf, and N. Maiden. Using Mobile RE Tools to Give End-Users Their Own Voice. In *18th IEEE International Requirements Engineering Conference (RE)*, 2010, pp. 37–46.

[42] G. Sindre and A.L. Opdahl. Eliciting security requirements with misuse cases. *Requirements Engineering*, 10 (1). 34-44.

[43] D. J. Solove. Understanding Privacy. *Harvard University Press*, London, 2008.

[44] A. Sutcliffe, S. Fickas, and M. M. Sohlberg. Personal and contextual requirements engineering. In *Proceedings of 13th IEEE International Conference on Requirements Engineering*, 2005, pp. 19–28.

[45] A. Sutcliffe, S. Fickas, and M. Sohlberg. PC-RE: a method for personal and contextual requirements engineering with some experience. *Requirements Engineering*, vol. 11, no. 3, pp. 157–173, 2006.

[46] D. R. Thomas. A general inductive approach for analyzing qualitative evaluation data. *American Journal of Evaluation*, 27 (2). 237-246.

[47] J. Y. Tsai, P. Kelley, P. Drielsma, L. F. Cranor, J. Hong, and N. Sadeh. Who's viewed you?: the impact of feedback in a mobile location-sharing application. In *Proceedings of the 27th international conference on Human factors in computing systems*, Boston, MA, USA, 2009, pp. 2003–2012.

[48] T. T. Tun, A. K. Bandara, B. A. Price, Y. Yu, C. Haley, I. Omoronyia, and B. Nuseibeh. Privacy arguments: analysing selective disclosure requirements for mobile applications. In *20th IEEE International Requirements Engineering Conference*, Chicago, Illinois, 2012.

[49] R. K. Yin. Case study research: Design and methods. *Sage Publications*, 2013.

[50] E. Yu and L. M. Cysneiros. Designing for Privacy and Other Competing Requirements. In *2nd Symposium on Requirements Engineering for Information Security (SREIS'02)*, Raleigh, North Carolina, 2002.

[51] Y. Yu, T. T, Tun, A. Tedeschi, V. N. L. Franqueira, and B. Nuseibeh. OpenArgue: Supporting argumentation to evolve secure software systems. In *19th IEEE International Requirements Engineering Conference (RE)*, 2011, pp. 351–352.

[52] C. Zins. Conceptual approaches for defining data, information, and knowledge. *Journal of the American Society for Information Science and Technology*, vol. 58, no. 4, pp. 479–493, 2007.