International Journal of Smart Home
Vol.3, No.2, April, 2009

Personal Privacy Management for Common Users

Susana Alcalde Bagüés, Luis A. Ramon Surutusa, Mikel Arias,
Carlos Fernández-Valdivielso and Ignacio R. Matías
Public University of Navarra
Department of Electrical and Electronic Engineering
Navarra, Spain
salcalde@gmail.com, surutusa@gmail.com, mikelarias@gmail.com,
carlos.fernandez@unavarra.es, natxo@unavarra.es

Abstract

In this work, we introduce the Privacy Manager, a user interface designed to allow non-expert users to manage privacy in the envisioned era of pervasive computing. The Privacy Manager is part of the implementation of the User-centric Privacy Framework, which was introduced as a novel mechanism to enable personal privacy for the inhabitants of the smart home. The Privacy Manager interface incorporates a set of application parts designed especially to meet the requirements of user friendliness, and privacy awareness, with the goal of making privacy management an affordable task for common users. Our first prototype allows to: i) customize permissions for the disclosure of their personal data, ii) control active and passive interactions with services, iii) define obligations to be negotiated on the usage of the data, upon transmission, iv) be aware of privacy related issues such as granted and denied permissions, v) apply alternative privacy mechanisms to access control, as white lying and obfuscation, vi) adhere to enterprise privacy policies based on a contractual relationship with an enterprise or organization. Providing people with tools to control their privacy is critical to guarantee the success of pervasive computing.

1. Introduction

Privacy is a prime concern in today's information society and one of the most challenging topics to consider when designing pervasive computing spaces, characterized by its ubiquitous intelligence and personalized services. Pervasive computing is driven by the idea of computers becoming invisible, embedded into everyday objects and seamlessly interconnected, to provide users with personalized services and information in an anywhere, anytime fashion. The outcome of pervasive computing seems to be the digitalization of our lives, to allow computer systems to automatically process them. It does not come as a surprise that pervasive computing has the potential to change our perception of privacy in an even more significant manner than the Internet did.

In our work, we addressed the idea of extending privacy control in a substantial matter toward holistic privacy management, with the collaboration of *personal privacy* and *enterprise privacy* enforcement systems. Our claim is that it is not sufficient to rely only on enterprise's willingness to support individual's privacy requirements, as it is typically deemed sufficient today. Protecting individuals' privacy in pervasive computing requires an additional level of privacy protection, which we called *personal privacy enforcement*; together with the enforcement of enterprise's privacy statements,

government policies and privacy laws, it is necessary to deploy mechanisms specifically for defining, communicating and enforcing people's privacy preferences. Privacy is a very subjective concept, what is acceptable to one person, may be unacceptable to another. As Langheinrich said "*protecting people's privacy is a very personal affair. Something that cannot be solved without taking people's habits, preferences, and moral views into account*" 0. For this, we have developed the User-centric Privacy Framework (UCPF) 0 to act as the *trusted privacy manager* of a small and controlled number of users, e.g. inhabitants of the smart home, and assist them in interactions with pervasive services.

We particularly address in this paper how the inhabitants of a smart home can be empowered to decide on the exchange of personal data on a much simpler, automatic, and fine-grained level than possible today to prevent them from losing their privacy to enterprises and data sellers in the upcoming era of pervasive computing. We present our solution, the *Privacy Manager* interface as the UCPF's GUI, which allows users to specify and communicate their privacy preferences, thus, to manage privacy by themselves. Simplicity, user friendliness and awareness were the three key concepts that guided its design. The Privacy Manager interface presents a new set of tools designed to tackle novel aspects of personal privacy management. With our interface a user is able not only to define his own privacy policies but also to specify the obligations that should be negotiated with the service recipient, on secondary use of the data, or set a *Virtual Context* to be distributed instead of the real data, or even configure different levels of obfuscation, called *Transformations*, to limit the granularity of his location.

Our first prototype includes six applications: "Customize your Permissions", "Customize your Services", "Organization Policies", "White Lying", "Obligations" and "Privacy State". They allow now users managing privacy but at the same time avoiding overwhelming them with the troublesome task of creating and administrating their privacy preferences.

2. Scenario

The following example scenario illustrates the future privacy management needs of users of pervasive systems. It shows the richness of restrictions that users of the new generation of context aware mobile services (CAMS) might want to apply to control the distribution of their personal information. Figure 1 summarizes those restrictions based on the examples used in the scenario presented below. Moreover, the scenario provides a general overview of the of privacy issues addressed by our privacy framework, the UCPF.

Ivan lives in a smart house with his wife Maria and his two daughters. The house is designed for making life more comfortable and secure. The UCPF system has been installed, in the residential gateway, and all the family members can now store their privacy preferences and administrate their own personal privacy.

Ivan travels frequently but he does not trouble anymore with planning his trips. Now, he is subscribed to the "Journey Planner" service. Thus, after setting up his personal and the company preferences and keeping up to date his travelling calendar, the "Journey Planner" arranges automatically all his trips. Ivan relies on the UCPF to disclose his calendar appointments filtered; only those appointments with event type "travel" are revealed. Moreover and for security reasons Ivan's employer does not allow to reveal travel information related to projects classified as "high security" and

the UCPF needs also to evaluate the organization policy that checks if the topic of Ivan's appointments are set up as "high security" before distributing the information.

Ivan has met lot of people all around the world and he likes to keep in touch with many of them, networking is important for Ivan's work. Ivan uses one of the known "Friend Finder" services to track his contacts. He likes to get alerts about friends that are nearby. However, Ivan only wants to reveal his position when he is not working and to people that is in the same city as him.

One day, after Ivan has agreed on meeting that evening with his friend Bob, who is that week, surprisingly, sleeping in his same hotel; everything seems to be working against him and he needs to work until late. Tired and disappointed after a bad day, Ivan leaves the customer's office for working in his hotel room. On his way to the main entrance he remembers his appointment with Bob. Not willing to give long explanations, Ivan switches to a white lying state and sends Bob a SMS to cancel their dinner. His privacy framework will maintain his location as "by client" and his situation "working" to all his friends until 2am the next morning and will not have him being disturbed.

Furthermore, Ivan requires from all services using his activity and location, that they do not store his data or, if yes, to delete them within one week or a maximum of one month and he wants also to get a notification if a service discloses his location to the same recipient more than 5 times a day.

Finally, Ivan uses a "Restaurant Finder" application to request for known restaurants when he is out in a new city, with this application he can automatically see the restaurants registered in the area where he is located. The UCPF always discloses his location to the restaurant finder with accuracy area, without revealing his exact position.

Service	Subject	User Context	Subject Context	Resource	Disclosure Level	Service Obligation	Virtual Context
Journey Planner		Calendar event: "travel"		Calendar appointment			
Journey Planner		Calendar topic: "no high security"		Calendar appointment			
Friend - Finder	Group: Friends	Activity: "no working"	In same city	Location and activity	Granularity Street	- Delete data in max 1M - Notify if more than 5 accesses same person	
Friend - Finder	Group: Friends			Location and activity			Loc: "by client" Act: "working"
Restaurant Finder				Location	Granularity Area	- Delete data in max 1M - Notify if more than 5 accesses same person	

Figure 1. Scenario's Privacy Restrictions

The scenario reflects that contrary to traditional access control, which is primarily based on identities and static attributes known a-priori to the system 0, the enforcement of personal privacy in pervasive computing shall involve the evaluation of restrictions on dynamic contextual information, such as the location or activity of the user, as well as the establishment of obligations on the secondary use and obfuscation of data before granting its collection. We have also introduced here the concept of using *white lies* 0 in order to allow the disclosure of a generated *virtual context* when a user wishes to

restrict the disclosure of his real data to a selected recipient. Our privacy framework was designed to provide the means to fulfill such restrictions and integrate complementary privacy protection measures to cover the new privacy needs raised by the development of the pervasive computing vision and a continuous collection of context data.

An important part of our work was focused on studying how common users can understand and thereby manage the different privacy protection tools implemented in the UCPF. In this paper, we discuss our solution developed to help users of the smart home manage their privacy by making use of the set of restrictions shown in Figure 1. As a first step, we evaluated the following questions: *How can a user understand the concept of and define personal privacy policies? How can a user specify restrictions on context data or even on the recipient context on his own? How may a user still apply plausible deniability in pervasive computing? How may he check on those organization policies assumed? How can our user negotiate obligations on secondary use or set up the granularity level of information to be disclosed?* In a second step we designed a set of applications to address these questions and meet our requirements regarding user privacy protection and manageability. The third step will be to test our user interface, the Privacy Manager, with a set of targeted users and evaluate its acceptance and usability based on our initial questions, which is currently part of our ongoing work.

3. Background and Analysis

Before going further, we introduce in this section relevant aspects related to the design and implementation of the UCPF, base for the specification of the Privacy Manager interface presented in this paper. The UCPF architecture has been designed according to a set of requirements defined to reach, what we consider, the "right level of personal privacy"; it is not possible anymore to keep one's person life total private but it is possible to deploy mechanisms to ensure that data is shared to the extend people want and not more. Figure 2 shows those requirements as grouped into three subsets. During their definition our goal has been to maintain the balance between *Privacy Protection* and two other key principles, namely: *Service Usability*, and *User Manageability*. We want to provide non-expert users with the means to control their personal privacy but without forgetting the main reason for all this; we need privacy protection because we want to use some of the offered services in exchange for revealing personal data. Privacy protection obviously must not impede the use of CAMS. And of course we need to empower non-expert users to manage privacy by their own. User Manageability involves, among others, to limit the number of parameters that a user needs to configure. A mistake would be to overwhelm users with the burdensome task of creating and managing privacy.

The requirements listed under the Service Usability principle do not affect the design of the UCPF's GUI, thus, we focus our discussion on the set of requirements grouped under the *Privacy Protection* and *User Manageability* principles.

i) *Privacy Protection*: the realization of this principle involves five design requirements to be implemented.

- **User-centric**: Individuals need mechanisms to define under which circumstances data can be disclosed. In order to offer a controlled distribution of sensitive personal data and spare individuals from spending time on setting their privacy

preferences for each encountered service separately, the collection and distribution of users' personal data should be centralized. Therefore, we have designed a single interface to allow inhabitants to configure their own privacy preferences and deal with inside and outside services of the home environment.

- **Privacy-aware Access Control**: As a corollary to the previous one, this requirement states that the system shall provide appropriate access control mechanisms for allowing the specification and enforcement of user privacy policies during collection and thereby users needs an application to define the "when, what, how, and who" of accessing personal data.
- **Context Awareness**: Recent studies on the perception of privacy 0 by individuals concluded that user preferences vary depending on place and social context. Thus, privacy policies should be made context-aware. In addition to the typical restrictions on the recipient or on the purpose of the data collection, dynamic constraints related to a user's environmental context should be possible. Most attributes that describe an individual and the environment are dynamic context, e.g. location, time, temperature, blood pressure, activity, etc. In our first prototype we focus on the specification of constraints based on the user location and activity, as well as on the recipient location and activity.
- **Trustworthiness**: Privacy enforcement needs to trust the involved parties in that they will fulfill their duties with respect to privacy protection. This requirement is twofold: on the one hand, it means that a user must trust the entity in charge of enforcing his privacy preferences, thus, under the motto "no better place than home" the residential gateway provides the ideal environment to deploy our UCPF. On the other hand, users need to trust that once the data is disclosed, e.g. after enforcing a positive policy specified by a user, data is still treated following his specifications. This is done in the UCPF with the definition, negotiation and monitoring of *Obligations* on secondary use of the data.
- **Disclosure Level Control**: Data obfuscation measures should be provided to control the granularity of the information to be disclosed, to allow fine-grained control over the quality of data transmitted. In the UCPF we have implemented a set of processes, called *Transformations*, which allow setting up the level of granularity desired. In our current prototype for coordinates, civil address, calendar appointments and activity information.

ii) *User Manageability*: the realization of this principle involves three design requirements to be implemented.

- **User Friendliness**: There is a strong requirement for a common, easy-to-use interface for giving a user the possibility of managing his privacy without overwhelming them. This is a common requirement often mentioned, although at present no implementation is available and also reference work such as The Faces Metaphor 0 was discontinued.
- **Awareness of Privacy Status**: Users need to be aware of personal privacy issues to properly administrate privacy. The increase of information flow back to the data owners regarding past interaction and service privacy criteria will help users avoid risky situations.
- **White Lying**: By definition, pervasive computing environments are supposed to be largely automated and "always on". In a certain sense it follows that people do not have the possibility to "switch-off" the system. We introduce the requirement

of adapting the concept of white lies as a way to "disconnect" individuals temporarily from a pervasive computing environment in a plausible way and maintain standard social interaction patterns, our approach to the integration of white lies is described in 0.

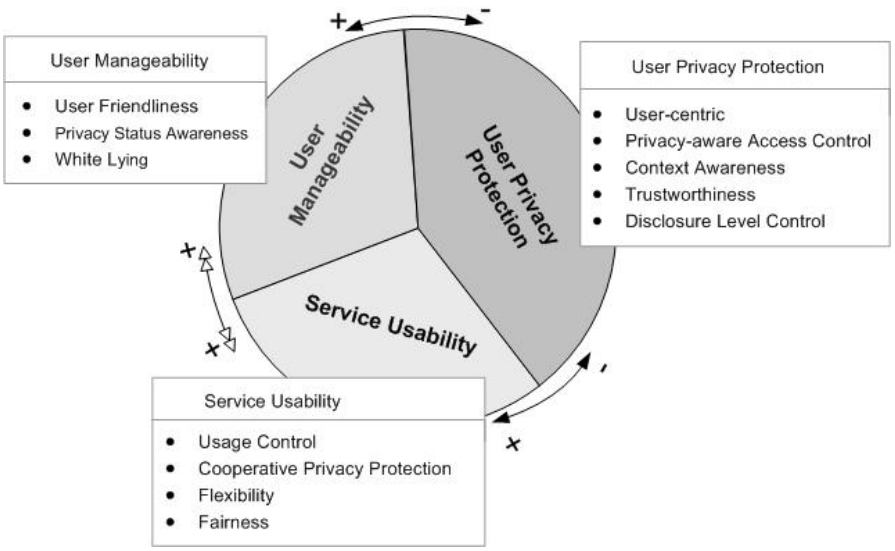

Figure 2. Requirements for Personal Privacy

As we mentioned, the first step in the design of the Privacy Manger interface was the evaluation of a set of questions, questions raised by the requirements described in this section. The implementation of these requirements in the UCPF together with our goal of enabling users of the smart home to manage their own privacy led us to the development of six applications, namely: "Customize your Permissions", "Customize your Services", "Privacy State", "White Lying", "Obligations" and "Organization Policies".

The application "Customize your Permissions" was defined to meet the requirements of *Privacy-aware Access Control* and *Context Awareness*. It allows the speciation of positive permissions and guides users in the definition of constraints on their and the recipient (subject) context information. The next application called "Customize your Services" can be used to set the transformation process (obfuscation level) to be carried on per service and resource, e.g. in our scenario to set the granularity of the information to be disclosed to the "Restaurant Finder" to level area. Thus, it addresses the *Disclosure Level Control* requirement. The "Privacy State" application, still under development, is designed to fulfill the requirement of *Privacy Status Awareness* and to provide users with information regarding data consumer privacy policies, state of obligations and active data collectors. The fourth application, "White Lying", is developed to meet the requirement of *White Lying* and allow users to set a white lying state. The application "Obligations" allows the specification of the set of obligations that need to be negotiated with a third-party service in order to meet the requirement of *Trustworthiness*. Obligations are requirements that must be agreed on by obligation subjects before authorizing the access to the data. Obligations can be seen as a binding statement to take some course of action in the future by the obligation subject, the service recipient of the data 0 in our scenario two examples of obligations are given, to delete

de data within a period of type and to notify after more than five accesses by the same person. Obligations can be also used to restrict allowed purposes. Finally, the application "Organization Policies" enables users to import, check and accept or deny policies defined by an organization, for instance, the rule defined by Ivan's employer to avoid the disclosure of calendar appointments tagged as "high security".

4. Interface Design

The Privacy Manager is designed to allow users of the smart home easily to specify their privacy criteria, concerning collection of data, desired obfuscation level and obligations to be negotiated, as well as to provide intuitive interfaces for awareness and feedback. The core idea is to empower non-expert users to manage their own privacy preferences with respect to the different tools provided by the UCPF and thus to meet the requirements specified for personal privacy enforcement. But due to the intrinsic complexity of semantic languages (used in our framework in the specification of policies and context data [ref SET]), plus the lack of general knowledge of individuals of how to deal with privacy issues and protect their privacy, the task of designing the appropriate user interfaces is not trivial.

Obviously nowadays, almost everybody uses computers in his daily life at home, in the office, even in cash points. People know how to manage information using GUI (Graphical User Interfaces) and keyboards and mice, obviously. Therefore, a GUI also seems to be the best solution to deal with privacy management tasks. We have developed the Privacy Manager interface, the UCPF's GUI, based on conventional window-based control elements such as list, buttons, etc. Our goal was not to develop a new graphical environment to manage privacy, as others before have tried. For example, in the case of the "Virtual City" application a city's map is used as interface for managing identity-related processes. In 0 studies showed that most users still preferred to use conventional interfaces, 24 out of 34 participants selected a "traditional" browser interface rather than the "Virtual City". It is our believe that users need to learn how to manage privacy prior to introducing "fancy" interfaces, it is more important to provide easy to understand concepts for describing and managing privacy, which can be in turn be accessed by different GUIs (e.g. Virtual City) later.

Nevertheless, the idea of building user friendly and easy-to-use interfaces has been followed by designers since the beginning. In fact, there exists an area dedicated to human-computer interaction that, since 1960, studies the requirements needed for designing high quality human-computer interfaces 0. In 0 the main design guidelines for GUIs are collected. Below, we list those we emphasized in the implementation of the Privacy Manager Interface.

In advance: A good application design means that the application "thinks in advance"; it is able to anticipate what a user may want to do next, and display the information and tools he may need. In the same context, it is recommended to show default values, when possible, to guide the user in the process of inserting data 0. The Privacy Manager has been developed following this criterion, displaying tools only when needed and providing different sets of default setups, from policies and rules to simple parameters.

Autonomy: It is a good and recommended practice to enable free exploration of the interface, allowing users to navigate though the different applications and windows, and to discover available options and tools. In other words, users should be able to "click" around without getting into trouble. This makes users feel comfortable with the GUI, and in general

reduces initial learning. In the Privacy Manager we always keep the logout and home button visible, to allow users to go back or exit the application at any moment. The application also is provided with information and question messages that inform users about the consequences of pressing a certain button.

Consistency: This concept is related to the idea that the same thing must be done following a similar way in different windows or applications, e.g. exit an application. It is important to keep a similar look and feel to avoid pushing user to figure out how to do something that they usually already know. In the Privacy Manager we implemented known features such as switching from one field to other with the “tab key”, accepting changes by pressing “enter”, accessing the menu always on the top left hand side of the window, or redirecting to the application home window by pressing the icon “house”, etc.

User effectiveness: An ultimate goal of any user interface it is to improve the performance of its users in the process of realizing a particular task. It is especially important that an application avoids losing information, due to potential user mistakes, or unavoidable failures such as power failure. In the Privacy Manager there is no real danger of losing important information. The process of adding a new rule into the system, involves a maximum of 6 quick and consecutive steps. As the rule is saved at the end, a user could only lose the information of that last rule, which is easy to check and resolve. Furthermore, useful tools for speeding up tasks have been included; a user is able to establish multiple preferences in one step, just by selecting some extra options rather than repeating tasks once and again.

Latency reduction: Another good design principle is to have heavy processes running in the background, e.g., in a separate thread. The idea is to keep the interface always non-blocking and ready to interact with its users. Otherwise, a user could get frustrated and decide not to use the application anymore, if it is constantly busy executing internal tasks. In those situations where we cannot avoid that the application is busy finishing a task, e.g. if the task takes between half a second and two seconds, it is recommended to use an animated pointer. On the other hand, for those tasks that take longer than two seconds, the best solution is to add a progress bar. These measures have been implemented in the Privacy Manager.

Initial learning: The easier it is to use an application, the better it appears to users. Nowadays, there exist lots of alternative applications for performing the same task. Thus, users are ready to reject a GUI just because they feel tired or frustrated during a first try. Although there is not an alternative solution to the Privacy Manager, it is important always to favor initial learning. Because of that, as part of our ongoing work, we plan to test the usability of the application with a selection of potential users, and evaluate the initial learning time of the interface. There exists a trade-off between user friendliness and flexibility and a compromise must be found to favor initial learning. In the design of the Privacy Manager the number of restrictions that a user could apply, in the definition of a rule, was limited to those we thought are more common and easier to apply with real applications. As a result, the use of the interface limits the expressiveness of the policy language. Nevertheless, we are aware that due to the novelty of the concept developed, the use of the interface may require some extra time to learn how to manage it.

5. Interface Implementation

The Privacy Manager interface incorporates a set of application parts designed especially to meet the requirements and design guidelines described in this paper and make privacy

management an affordable task for non-expert users. Summing up, our first prototype allows to: i) customize permissions for the disclosure of their personal data, ii) customize the level of disclosure for each service - resource, iii) define obligations to be negotiated on the usage of the data upon transmission, iv) be aware of privacy related issues such as granted and denied permissions, v) apply alternative privacy mechanisms to access control, as white lying and obfuscation, vi) adhere to enterprise privacy policies based on a contractual relationship with an enterprise or organization.

The Privacy Manager interface was designed to be used in a domestic environment, e.g. by a small and controlled number of users. The six proposed applications are hosted by an OSGi framework and connected remotely with the offered UCPF services through the Sentry Manager Interface. The Sentry Manager Interface is implemented as an API used to access to the Context Handler, Obligation Manager and Noise Module UCPF's components, all of them deployed as bundles within the OSGI environment. Through this API it is possible to generate, upgrade or delete privacy policies, receive information about the current applicable policies, specify obligations and get feedback on potential privacy risks.

Figure 3. Access and Home Window

The first prototype provides a single interface to access all the applications. Part of our ongoing work is to develop a second version of the GUI, implemented as a web application, to promote mobility. The access screen, shown on the left hand side in Figure 3, is the first window that users find after starting the Privacy Manager interface. This window is used to login but also to register a new user of the system. Once a user is registered, he or she can access the offered applications, by just introducing "username" and "password". The button "Login" takes a user to the next window, the "home" window.

The "home" window, shown on the right hand side in Figure 3, displays six different options to manage personal privacy related to each of the applications mentioned. Two of them, the "White Lying" and "Organization Policies" are already designed but under development. The other four: "Customizing your Permissions", "Customizing your Services", "Privacy State", and "Obligations" are already implemented. Each of the applications is presented including a picture and a brief explanation to help users understand the functionality of the application. In this window, a user can easily discover how to access an

application by just moving the mouse around. Then, the selection area, extended to the whole panel, is highlighted with a blue color. The “home” window offers the possibility of logging off and also removing the user from the system by clicking on the “trash” icon.

5.1. Customizing your Permissions

The first selectable option, depicted at the upper left corner of the “home” screen, is the application that allows users to manage positive permissions. A special feature of the “Customize your Permissions” application is that it provides a default configuration for all its users, even if a user does not introduce any rule (permission) at all, the user is still protected, the system returns always a “deny” value, which applies if there is no a positive permission. With this application a user can add new permissions and check, update and delete those previously created. It also provides a special tool to manage contact groups, which can be used to group individuals into roles.

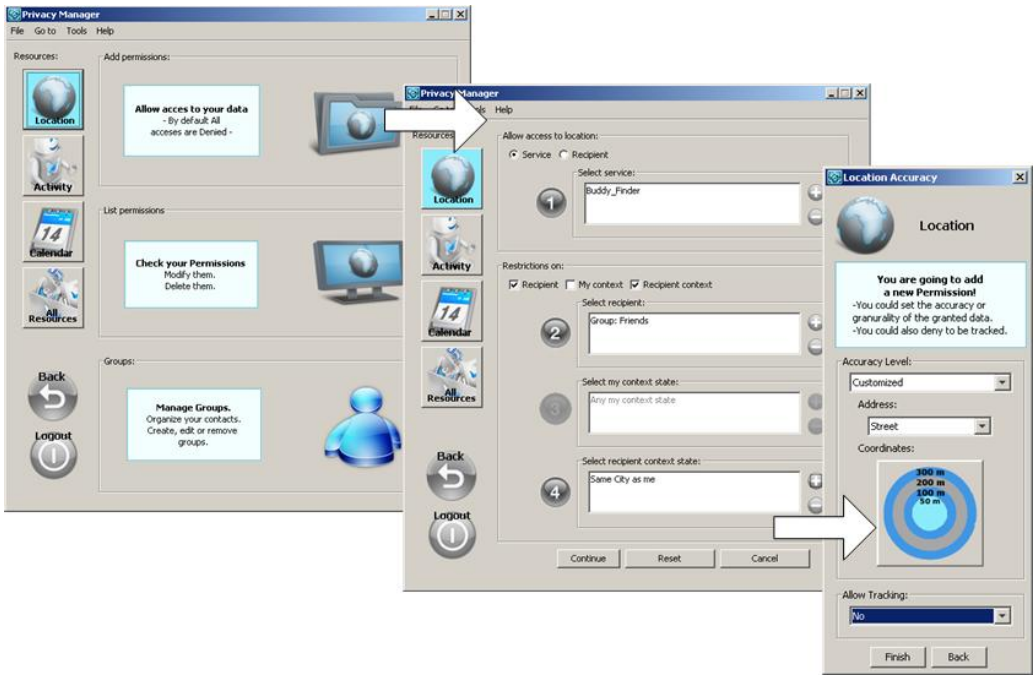

Figure 4. Customize Permissions Application, main screens

Figure 4 shows the main windows of the application “Customize Permissions”, in order to start adding a new rule the user has to select a resource button among location, activity, and calendar or select all resources, and press the “folder” icon shown in the first screen on the left hand side of Figure 4. After that a new screen opens, the one indicated with an arrow, to allow the definition of the restriction of that rule. First of all, the user needs to select from a list of registered services to which of them the positive permission should apply, then he can use the selection screens shown in Figure 5 to add restrictions on the disclosure of the selected resource. A user can define three types of restrictions to constraint the recipient of the data (person, group, organization). He can restrict the disclosure of his data depending on his current activity or/and location and also the location or/and the activity of the recipient.

After selecting the desired constraints and by pressing the button “Continue” the next screen is opened, which is used to set up the quality of the data to be granted and to allow or disallow a tracking action, before storing the new permission in the policy repository. The application offers also the possibility to specify the rule for a particular recipient without setting the service, for this the user needs to select in the center screen of Figure 4 within the “Allow access” frame the “Recipient” tag instead of the “Service” tag.

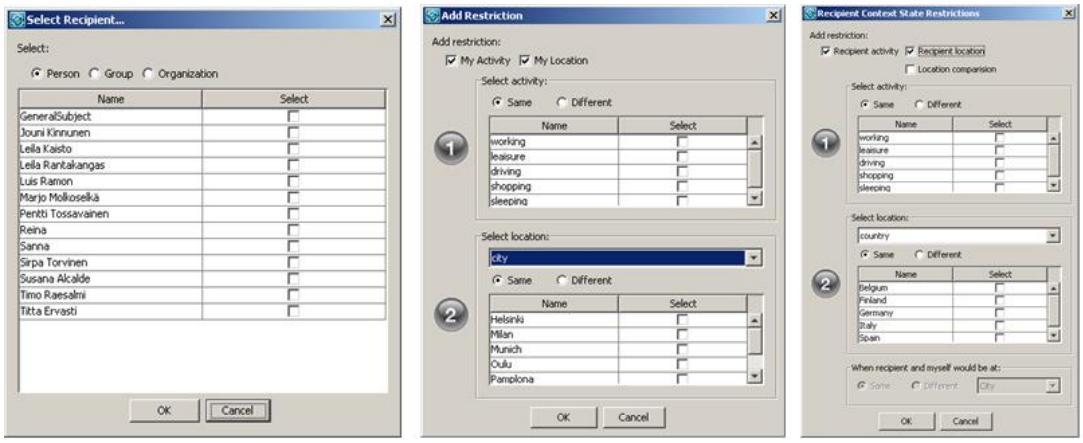

Figure 5. Customize Permissions Application, selection screens

5.2. Customizing your Services

The second option to create personal policies is the application “Customize your Services”, depicted at the upper right corner of the home screen. This application simplifies the process of adding new rules into the system for active interactions, when the user is requesting the service, since users just need to configure the accuracy of resources for each service. Theses rules do not include constraints, due to the fact that the user is requesting the service, thus, he should grant the access to his context in order to favor service usability. If one of these rules is enforced, the service always gets grant permission; the rule only controls the transformation, which sets the desired accuracy of the data to be disclosed.

The “Customize Services” application is displayed in Figure 6, once a user selects a service from the list or registered services, the resources used by that service, one or more of the four resources considered in our implementation (coordinates, civil address, activity, and calendar appointments) are activated. The user can change any previous configuration by selecting one of the predefined levels of granularity. The quality of the data disclosed is related to the transformation performed before releasing the resource. We implemented 5 transformations of type coordinates with the possibility to reveal coordinates with 50m, 100m, 150m, 200m or 300m maximum accuracy, 6 of type civil address, with them the user can select the disclosure level to “room”, “building”, “street”, “area”, “city”, or “country”. Other 3 of type activity, they allow to disclose either working or leisure activities, or to disclose only the activity type. There are 2 transformations more for calendar appointments, to filter work from personal entries. The difference between the 2 screen shots shown in Figure 6 is the number of resources activated, the Buddy Finder service uses the four resources while the Restaurant Finder only two of them, coordinates and civil address.

Figure 6. Customize Services Application

5.3. Privacy State

The application "Privacy State" was created to meet the system requirement of awareness of users' privacy status. The application should displays all the information related to privacy issues. In the current version, it lists granted and denied permissions sorted by date, and allow checking the rule responsible of such a granted permission as is shown in Figure 7. In the future, it should provide functionalities to track existing agreements on obligation sets, check the state of unfulfilled obligations, and monitor notifications sent to and received from a service. We would like also to extend this application by incorporating alarms on potential privacy risk and the possibility to visualize service side privacy policies.

5.4. White Lying

Intrinsically, pervasive computing environments are supposed to be largely automated and "always on". In a certain sense, it follows that people do not have the power to "switch-off" the system. We introduced the requirement of adapting the concept of white lying as a way to "disconnect" individuals temporarily from a pervasive computing environment in a plausible way and maintain standard social interaction patterns, further details are given in 0.

The White lying application is the user front-end application used to set a white lying state in the UCPF and allow the use of mechanisms for plausible deniability in pervasive computing. The application proposes for a selected recipient (recipient of a *white lie*) a virtual context (location-activity) to be disclosed instead of the real one. A virtual context is compiled based on the user and recipient visibility respect to each other, on existing permissions, the location of the recipient, the present and future locations of the user, and a set of parameters configured by the user. The process of setting a white lying state involves three consecutive steps, thus the application being implemented has consequently three main screens. The first screen is used to select the recipient of the white lie and triggers the process that evaluates whether a white lie for the selected recipient is allowed at all, the second screen informs the user of the visibility settings and services affected by such white lie state, the

third screen proposes a virtual context and allows a user to accept such white lie or select other manually.

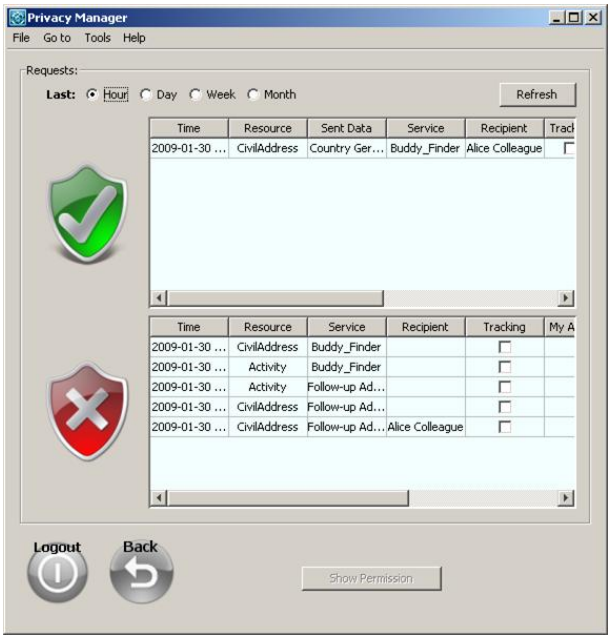

Figure 7. Privacy State Application

5.5. Obligations

Obligations are used to create automatic bindings with CAMS and ensure that data protection requirements are adhered to. Obligations in the UCPF have the following purposes: i) they specify actions that should be performed by a service, acting as the recipient of a user data, ii) obligations are used to automatically exchange user's preferences on secondary use with enterprises, iii) they enable the exchange of notifications between the UCPF and third-party services and with that post-disclosure life-cycle control. We created a set of predefined obligations and classified them into *system obligations* and *negotiable obligations*. The goals of system obligations are to control disclosures to third-parties, monitor changes on a service privacy policy and enable the access to collected data and data logs. They are mandatory obligations which are by default established independently of a user's preferences and beforehand of any commercial transaction with a service. Negotiable obligations, on the other hand, are selected by a user and negotiated with the service. Users can specify negotiable obligations to be agreed on during the evaluation of a service request before granting the access to the data. They represent the privacy constraints that a user may impose on an enterprise service when data is disclosed. Negotiable obligations are used to control the information disclosed among individuals (users of a particular service), authorized purposes, number of accesses and retention time of user resources. More details about obligations in the UCPF can be found in 0.

We have included the application "Obligations" to help users select obligations that should be negotiated with each service, before releasing any resource. In that application, negotiable obligations are grouped in sets of three obligations, optimal, acceptable and minimum, one for each of the three negotiation rounds allowed, which is reflected in the UI, accordingly.

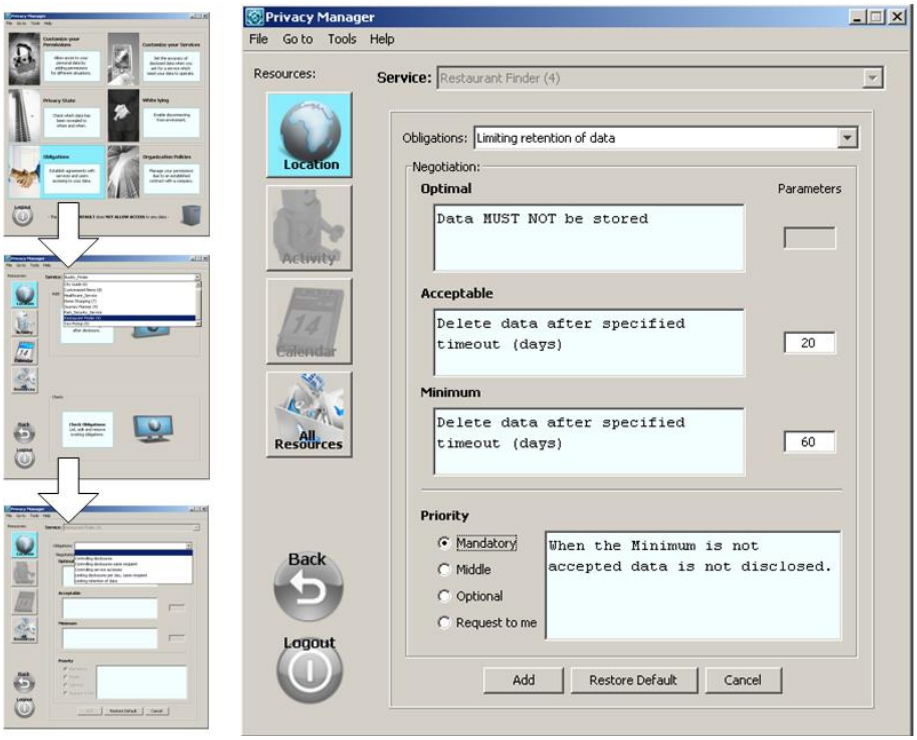

Figure 8. Obligation Application

Figure 8 shows a screen shot of our current application prototype for managing sets of obligations. A user can access this application directly from the "home" window by clicking into the panel "Obligations". The access window that is opened now, shown in the left hand side in Figure 8 guides users in the selection of a service provider and resource among location, activity and calendar, before allowing the access to the main screen of the obligations application. In the main screen, shown on the right hand side in Figure 8, the predefined sets of obligations are listed; a user can select the obligation set that he wants to negotiate with the service before granting the access to the resource. That window allows users to change the parameters of obligations belonging to a set and to select the priority of these obligations. A set always consists of the three mandatory obligations, labeled as "Optimum", "Acceptable", and "Minimum". They can be predefined and be re-used, obviously, and do not have to be defined separately each time. The application offers also the possibility of checking existing obligations and editing or canceling them.

5.6. Organization Policies

Organization Policies are enforced exclusively during binding interactions, when interactions with a service are regulated with a binding contract between parts, a user and an organization. For instance, a home-care nurse might be required to disclose her location to her employer when her activity state is "working" to better organize her calendar. The management of these types of policies entails two complementary processes: i) first, an organization needs to create a new policy for one or more users of the UCPF. The constraints that an organization policy can include are the same as the ones presented in the "Customize

your Permissions" application. In this case, as an organization is not a user of the UCPF, it needs an external web application, offered by the Sentry Registry UCPF's component, to be used to specify such policy, ii) second, a user of a UCPF can use now the "Organization Policies" application and download the policy from the Sentry Registry. The application allows users to edit and check the constraints included in an organization policy, in the same way that they can check a personal rule as shown in Figure 9, and either accept or reject it.

There, by just selecting one the listed rules, shown on the right hand side in Figure 9, the rule is translated into a human-readable syntax and displayed below on the "Description" frame. If the user clicks the right button of the mouse he or she can choose between either editing or deleting the rule. Once an organization policy is accepted, it will be evaluated each time that a request from such organization is sent to the UCPF.

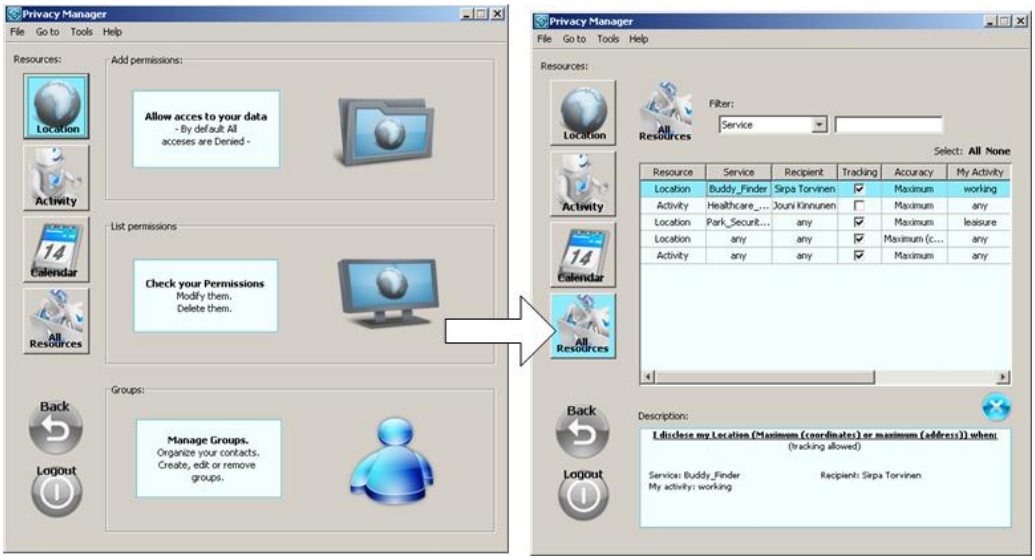

Figure 9. Edition and Deletion of rules

6. Related Work

To our knowledge there does not exist any other work in the area of privacy management that includes a set of applications as we presented here. We have developed six applications for helping users to configure easily their privacy preferences regarding positive permissions, with the addition of restrictions on dynamic context, disclosure level, obligations, white lying state, as well as providing tools for feedback. For instance, the Privacy Bird 0 presents a quite user-friendly interface that informs users if a web site provides a privacy policy and whether the user is protected or not, a user can easily see that just by checking the color of a small "bird". On the other hand, a user can only restrict the type of information to be disclosed and the purpose, it does not offer enough expressiveness and a user cannot add, for example, dynamic constraints or obligations.

Privacy management is concerned with how users may control everyday privacy in pervasive computing, assuming that regular individuals, voluntary disclose their contextual information to pervasive computing services. Lederer et al. emphasized in his work 0 also the

use of feedback together with control and developed a prototype named Faced Metaphor to manage privacy, which provides additionally to the enforcement of user faces preferences, real time feedback. In this system a user can assign a *face* to handle disclosures for each inquirer and for each situation the user might be in. For example, "*if a roommate makes a request while I am studying, show my anonymous face*". Feedback takes place in an integrated log of disclosures reviewable after disclosure; users can navigate their log to help them to control what information is flowing to whom and to know the *faces* used. The drawback of this approach is that users found difficulties remembering their own settings, regarding each of the *faces* created, and were not able to predict the result of any service interaction.

Other relevant work in the area of privacy management is the Privacy Mirrors 0, Nguyen and Nynatt introduced a framework for designing socio-technical systems for ubiquitous computing. The idea behind the Privacy Mirrors is to allow users to understand how they and their personal information is being sensed and used by pervasive computing systems. Some examples of their interfaces are the "People Counter", this prototype displays the number of persons in a room, showing that the systems is currently working and sensing the people entering in that room; The "Cartoon Parts" interface registers whether a smart home can identify its inhabitants by showing a picture and where that person is located; The "Calendar Mirror" interface combines in a display user's calendar with information about how that information has been accessed by others.

Other related research are "Semantic e-Wallet" 0 and "Houdini" 0. Both of them are concerned with ensuring that personal data is only made available to appropriate parties in appropriate contexts. Both have used rule engines to accomplish this. However, the first work does not discuss how users could express their privacy preferences by themselves. In "Houdini" framework, a user interface is presented and user preferences can be populated using templates. But it does not provide a way to customize obligations on secondary use, or accuracy of the information to be disclosed, or allow white lying. Houdini system makes use of user preferences to infer new permissions. Although, this could be seen as a way to reduce user settings and effort, it could also lead to undesired disclosures. User should be aware of their privacy affairs and take control over them. Our target, with the implementation of the Privacy Manager is to provide a single interface that allows users to enforce control and feedback, at the same time that we provide mechanisms to address the different aspects of privacy control, as are prevention, avoidance and detection 0.

7. Conclusions and Outlook

In this document, we presented our Privacy Manager interface, developed to allow non-expert users to interact with the User-Centric Privacy Framework (UCPF) 0 and define and administrate their privacy preferences, offering six different applications to control related aspects of personal privacy. In order to succeed, a balance between flexibility and easiness was met.

Part of our ongoing and future work is to complete the implementation of our first prototype and to realize an extensive field study, with a selection of 20 or more potential users. Thus evaluate different usability aspects of our Privacy Manager interface. Our main interest is to learn the difficulties that users might face managing personal privacy with our proposed applications.

References

[1] Marc Langheinrich. Privacy by design – principles of privacy-aware ubiquitous systems. In G.D. Abowd, B. Brumitt, and S. Shafer, editors, Ubicomp 2001 Proceedings, volume 2201 of Lecture Notes in Computer Science, pages 273–291. Springer, 2001.
[2] S. Alcalde Bagüés, A. Zeidler, C. Fernández Valdivielso, I. R. Matías, "Sentry@Home - Leveraging the smart home for privacy in pervasive computing", International Journal of Smart Home, 2007, vol. 1 No. 2.
[3] Jaehong Park and Ravi Sandhu. The UCON Usage Control Model. ACM Trans. Inf. Syst. Secur., 7(1):128–174, 2004.
[4] S. Alcalde Bagüés, A. Zeidler, C. Fernández Valdivielso, I. R. Matías, "Disappearing for a while – Using white lies in pervasive computing", In Proceedings of the 6th ACM workshop on Privacy in electronic society, Alexandria, Virginia, USA, 2007.
[5] Denise Anthony, Tristan Henderson, and David Kotz. Privacy in Location-Aware Computing Environments. IEEE Pervasive Computing, 6(4):64–72, 2007.
[6] Scott Lederer, Jennifer Mankoff, Anind K. Dey, and Christopher P. Beckmann. Managing Personal Information Disclosure in Ubiquitous Computing Environments. Technical Report CB-CSD-03-1257, University of California, Berkeley, 2003.
[7] Lalana Kagal. Rei ontology specifications, version 2.0. http://ebiquity.umbc.edu/ resource/html/ id/34/Rei-Specifications, July 2004.
[8] John Sören Pettersson Mike Bergmann, Martin Rost. Exploring the Feasibility of a Spatial User Interface Paradigm for Privacy-Enhancing Technology. pages 437–448, 2005.
[9] JCR Licklider and W. Clark, "On-Line Man Computer Communication", Proceedings of the Spring Joint Computer conference, San Francisco, California, 1962, vol. 21, pp. - 113-128.
[10] W. Hansen, "User Engineering Principles for Interactive Systems", Proceedings of the Fall Joint Computer Conference, AFIPS Press, 1971, pp. - 523-532.
[11] G. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", Psychological Rev., 1994, vol. 101, no. 2, pp. - 343-352.
[12] S. Alcalde Bagüés, J. Mitic, A. Zeidler, M. Tejada, C. Fernandez Valdivielso, I. R. Matias, "Obligations: Building a Bridge between Personal and Enterprise Privacy in Pervasive Computing", Proceedings of the 5th International Conference on Trust, Privacy and Security in Digital Business, Springer-Verlag, 2008.
[13] Lorrie Faith Cranor, Praveen Guduru, and Manjula Arjula. User Interfaces for Privacy Agents. ACM Trans. Comput.-Hum. Interact., 13(2):135–178, 2006.
[14] Scott Lederer, Jason I. Hong, Xiang Jiang, Anind K. Dey, James A. Landay, and Jennifer Mankoff. Towards everyday privacy for ubiquitous computing. Technical Report UCB/CSD-03-1283, EECS Department, University of California, Berkeley, October 2003.
[15] David H. Nguyen and Elizabeth D. Mynatt. Privacy Mirrors: Understanding and Shaping Socio-technical Ubiquitous Computing Systems. GVU Technical Report GIT-GVU-02-16, Georgia Institute of Technology, 2002.
[16] F. Gandon, N. Sadeh, "A Semantic e-Wallet to Reconcile Privacy and Context Awareness", Proc. 2nd Intl. Semantic Web Conference, 2003.
[17] R. Hull, B. Kumar, D. Lieuwen, P. Patel-Schneider, A. Sahuguet, S. Varadarajan, A. Vyas, "Enabling Context-Aware and Privacy-Conscious User Data Sharing", Proc. of the 2004 IEEE International Conference on Mobile Data Management, 2004.
[18] X. Jiang, I.J. Hong, and J.A Landay. Approximate Information Flows: Socially-based Modeling of Privacy in Ubiquitous Computing. In Proceedings of the 4th International Conference on Ubiquitous Computing, 2002.
[19] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving privacy in environments with location-based applications. IEEE Pervasive Computing, 2(1):56–64, 2003.

Authors

Susana Alcalde Bagüés

Received her MS in Electrical and Electronic Engineering from the Public University of Navarra, Pamplona (Spain). She is currently working on her PhD thesis in the area of Privacy in Pervasive Computing at Siemens AG, Corporate Research and Technologies, Software & Engineering. Germany.

Luis A. Ramon Surutusa

Received her MS (2008) in Electrical and Electronic Engineering from the Public University of Navarra, Pamplona (Spain). He is currently working at the Electrical and Electronic Engineering Department of the Public University of Navarra (Spain) in the area of sensor networks.

Mikel Arias

Received her MS (2009) in Electrical and Electronic Engineering from the Public University of Navarra. He is currently working in the IT Management Functions Division of the European Central Bank Germany.

Carlos Fernandez Valdivielso

Received his MS (1998) and PhD (2003) in Electrical and Electronic Engineering from the Public University of Navarra, Pamplona (Spain). He is currently an Associate Professor at the Electrical and Electronic Engineering of the Public University of Navarra (Spain) and also the Managing Director of Domotic Engineering. His research interest is in the areas of sensor networks and home automation.

Ignacio R. Matías

Received his MS (1992) and PhD (1996) in Electrical and Electronic Engineering from the Polytechnic University of Madrid (Spain). He is a Professor in the Electrical and Electronic Engineering of the Public University of Navarra, Spain. He has coauthored more than 250 book chapters, journal and conference papers related to sensors and home automation.