7

On Modeling of Ubiquitous Computing Environments featuring Privacy

Vivian C. Kalempa, Rodrigo Campiolo, Lucas Guardalben,
Urian K. Bardemaker, João Bosco M. Sobral
Federal University of Santa Catarina - UFSC
Computer Science Program - PPGCC
Distributed Mobile Computing and Network Security Research Group - DMC & NS
Brazil

1. Introduction

This chapter discusses a metamodel suitable for ubiquitous computing environments Weiser et al. (1999), Weiser (1993). It includes a way to pervasive computing and includes the aspects of mobility for resources and people. The pervasive computing explores the increasing integration of computing devices in our physical world, while mobility is studied in the context of mobile computing, which exploits the connectivity of the devices which move within the world of people.

However, there are still technical challenges that prevent ubiquitous computing be consolidated in people's lives. Currently, research has been done by focusing on technical matters, such as the connection of devices and the building of applications for these environments. Issues such as security and privacy are still poorly treated. In this article, the challenges to ensuring privacy in ubiquitous computing environments are explored. A metamodel that aims to several aspects of ubiquitous computing is extended to the aspects of privacy. For instance, the degree of anonymity provided by an environment may be achieved. The chapter approaches on privacy. This one is a right of every person and many nations have laws in their constitutions that guarantee to the citizen the right to possess it. However, privacy can not be guaranteed only by laws, especially when it comes to digital data. This problem has been tackled in conventional computing for some time and the solution that has been used is cryptography. This solution has been satisfactory for the current paradigm, the personal computing. The ubiquitous computing is a new paradigm where the environments have sensors and computing devices capable of computing and communication. The user can communicates with such environments through their personal devices and vice versa. In ubiquitous computing, privacy has achieved new dimensions, which were often idealized by books and movies, but in modern times are becoming reality.

This chapter presents and discusses the dimensions of privacy in the context of ubiquitous computing, the issues being addressed by the scientific community and provides a model for addressing some of these issues in environments closed. This model is then simulated through a simulator and a metric Diaz et al. (2002) is used to measure the degree of anonymity achieved.

In the follow, section 2 presents foudations and the methods. The section 3 describes

a metamodel for ubiquitous computing environments. The main issues surrounding the privacy (services and restrictions) in ubiquitous environments are describes in section 4. Section 5 presents an extension to the metamodel developed by Campiolo (2005) to describe ubiquitous environments, with features to ensure privacy in such environments. In section 6 is presented a case-study, which was developed using the metamodel proposed and which was simulated. Section 7 contains important conclusions of this the work.

2. Foundation and methods

In this section the base concepts and the methods used for developing the specification are shown. It describes the entities, requirements and features of the environments and the formal language used at the specification process as in Campiolo (2005).

2.1 Entities

Entity refers to all of the instances that somehow collaborate for the formation and definition of a ubiquitous computing environment. In the current context, entities are people, devices, softwares and communication medias. People engage to the environments through the relation with other entities and behaviors. Devices with communication and/or computing capacity are the foundation for the existence and the growing ascension of pervasive computing. Softwares provide mechanisms for programming and to control devices. Finally, the communication medias, in special, wireless communication, are responsible for establishing connectivity among all entities that compose ubiquitous computing environments.

2.2 Features

Features define properties and important requirements for composition of the environments and must be respected by elements and in the modeling process. The most important are:

- Invisibility: it is the disappearance of user perception on the technology used;
- Intelligent environments: presence of saturated environments with electronic devices and defined frontiers, with capacity of computing and communicating to itself and other devices that present to them;
- Context awareness: awareness of location of the devices in pervasive world to use information on managing and communications in the environment;
- Security and privacy: to assure security of information and of the physical devices and to assure privacy in an environment with constant interactivity and connectivity.

3. The elements of a metamodel

In this section the model got though formal specification of the elements and features of ubiquitous computing environments is shortly presented as in Campiolo (2005) and Campiolo et al. (2007). Modeling aspects are presented, detailed and discussed informally, through explaining sentences, and formally, through the model in Object-Z Spivey (1989) and Duke et al. (1991).

3.1 Common features

In Figure 1 several properties of the physical elements that compose the scenarios of a ubiquitous computing environment are common. By using the concept of inheritance, the

common features were extracted and aggregated within a single class. A *ModelBase* class defines common properties of four elements of the model: people, objects, entities and spaces. An identification property exists to represent uniquely a physical object whereas the class and description properties allow to specify object details. A position property defines the absolute spatial location of a physical entity in the scenario. Every element that has this feature can be located through coordinates. The emittedSignal property has the function to aggregate and represent the set of signals that an element can transmit.

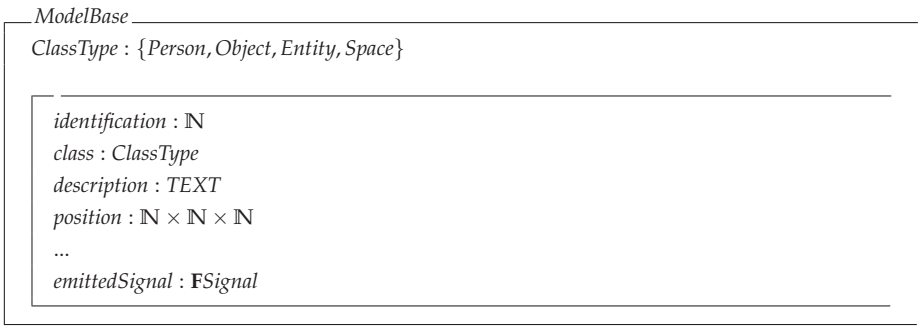


Fig. 1. Features of the *Model Base* Class

3.2 People

The whole essence of the representation of a person can be modeled for who is in a ubiquitous environment. A POSITIONAL type defines people positions in the environment and the characteristics that relate to position, such as velocity, direction and orientation. A PHYSICAL type represents physical characteristics, such as gender, age, size, among other. A PSYCHOLOGICAL type type represents abstract characteristics like emotions, interests, temper, at last, characteristics related to psychological.

3.3 Objects

Objects are all elements that do not have computing and communication, unless they are aggregated or embedded to them.

An initialState property is a set of state variables and defines the object initial state. The possibleState property is a set of state variables and defines all possible and valid values for composition of any object state. The changeState property is an association function among the events that change the object states with the transition relationship between the current values and the new values of the state variables changed by the event. Finally, the associations property allows the association of elements to objects.

3.4 Entities

An entity is the basis for the specifications of sensors, actuators and devices. These ones have common features and need a structure to restrict access to determined properties. The result is the creation of an abstract structure named Entity.

An enabled property defines if the entity is active or inactive. The connections, communication and channel properties specify with which entities the connections, protocols and physical communication channels are maintained.

3.5 Communication

Connectivity in ubiquitous computing environments is essential. Connectivity is provided by an *Entity* class which can be described. It defines for an entity the relation among the physical environment, protocol and connection. The physical mean is specified through a *CommunicationChannel* class, the protocol by the PROTOCOL basic type and the connection by a pointer to an entity. Despite an *Entity* and *CommunicationChannel* classes specify properties to represent and control communications, they are not enough to represent active communications and to a possible simulation management. To help on these issues, must exists a class named *CommunicationController.*

3.6 Events and signalings

Events and signalings in the model are represented by three classes: *Signal, Event and Command*. These classes represent respectively signals, events and commands. In this case, signals are information emitted by any element of the pervasive world; events are notifications issued by entities, in special, devices; and commands are orders and parameters to change object states. The common features of these classes are represented by *Trigger* base class. Common properties like event identification and content representation are provided by this class.

3.7 Location

Through the location information of people, objects and entities, environments manage resources and interactions among the elements present in the spaces. For the model, location information must support the features defined for real environments and even, provide practical mechanisms to manage this information. Absolute location of all elements is defined by the *ModelBase* class. The *SymbolicLocation* and *RelativeLocation* classes specify symbolic and relative location, respectively. They are aggregated by the *Location* class. As it is complex to maintain the location of the entities using location symbolic or relative, every element that uses that location type it should be registered in the location controller. To manage the location of all objects there is the LocationController class.

3.8 Situations

Occurrences or events generated in real world, or even in a simulation, are not all time-dependent. Some of them have relationship with time just by running in a random instant. Besides the existence of random events, there is a subset dependent on a collection of states of given elements. The term used in this paper for these events is situation-based events. Formally, they are event triggered when a finite event set is reached or activated and originate a determined situation. The *SituationController* class is a structure that aggregates the states of interest of a determined situation, associates and triggers events when states are activated.

4. On Ubiquitous environments

In ubiquitous computing, the technology is very close to people and lives in various scenarios that might be considered real. According to this paradigm, the computing elements should be invisible or to induce the minimum of distraction to the user. Based on this idea, is not acceptable that users are often interrupted with alerts and options to configure, accept or reject any type of intrusive action. In this section, we present services that can be intrusive in

ubiquitous computing environments, the issues to be considered and the restrictions that may be imposed on these services.

4.1 Services

Most services in ubiquitous computing are not yet widespread or applied in real environments Aoyama (2008). Below are described some of these services and their implications for privacy:

- **Service of product identification:** Through this service it is possible count and track a product that a consumer is buying. The implications are the lack of control of the types and use of information being collected and the possibility of an external entity track the contents of the purchase outside the shop;
- **Service of warning proximity:** This service informs when a known and registered person with your device is close to its physical location or in a common environment. The problem is that people do not always want to be located in certain situations or times;
- **Advertising service:** This service sends advertisements of products to user devices when they are close to their shops. In this case, there are some issues to be considered: (a) How to define policies to restrict the advertising? (b) How to avoid the stressful protocols to obtain this informations? (c) How to map the interests of a client? (d) How to define the limits to achieve this mapping?

4.2 Restrictions

The services introduced to ubiquitous computing aims to facilitate the implementation of tasks of the user. However, the services must comply with certain restrictions that do not become intrusive.

A classification for these restrictions is presented below Myles et al. (2003):

- **Temporal**: it determines the time periods in which the service is available or disabled. For example, a user does not like to be located in the time for lunch;
- **Localization**: this restricts access to informations or to the device based on the user's location. For example, in a restaurant the user can allow a service to obtain its name for a personalized treatment;
- **Organization**: it defines who and when a person can be located. For example, an employee of a company want to be found only when he is in the physical limits of the company;
- **Service**: this defines what services a client device can access. For example, when entering an environment with ad services, you can restrict what is allowed to access;
- **Order**: it defines what informations may be disclosed for a given service. For example, to complete a registration the client defines what data are relevant to be transmitted from your device to the register;
- **Situation**: this defines the situations in which policies defined for a service may be overlapping. This type of service requires a level of intelligence for the device. For example, a user does not want to access any service, while he is in the room with your boss;
- **Group**: it defines a common group that can access a set of user information. This restriction applies to devices from other users. For example, a user wants to share work information with all in their sector;

- **Interest**: this determines whether services or information transmitted to the device are of interest to the user. For instance, a user may wish to receive results of football matches, then he can set as interest this type of information.

5. Privacy model

This section presents an extension of the metamodel for ubiquitous environments created in Campiolo (2005) by presenting the aspects related to privacy Langheinrich (2001), Jiang & Landay (2002), Cheng et al. (2005), Bhaskar & Ahamed (2007). These aspects are presented and discussed informally, through explanatory sentences, and formally, through the model in Object-Z. In addition, the mathematical notations such as those that can be used in Object-Z are usually adopted, therefore accurately describe the properties of a computational system Duke et al. (1991).

In closed ubiquitous computing environments[1] the issue of privacy can be ensured internally by a local system. Thus, privacy violations and the problems caused by the communication must be protected in the environment.

Based on this assumption, in this research are considered the issues involving the environment and the individuals within the limits of that environment. Therefore, the interaction between devices of different individuals in the environment is not addressed.

One of the initial problems is about the user's device communication with devices of the environment. Additionally, all communication between devices consumes energy. Therefore it is necessary to avoid stressful protocols and repeated attempts at communication.

The metamodel presented in Campiolo (2005) does not allow specifying the problems relating to privacy on ubiquitous environments. The Figure 2 shows all classes built in Campiolo (2005) and within the red rectangle are created three classes that are appropriate and based on the concepts of (1) anonymity Pfitzmann & Köhntopp (2001), (2) the use of pseudonyms Beresford & Stajano (2004), (3) the user's preference profile Lederer et al. (2002) and (4) the creation of mixing zones Beresford (2005), if necessary the existence of these in the ubiquitous environment.

The *Service* class represents the services in ubiquitous environment and are detailed in Figure 3. These services are provided by devices and sensors and, as discussed in section 4.1. They can be intrusive and annoying that may pose serious threats to privacy of such environments.

The property *identification* uniquely identifies a service on the environment. The property *description* can provide some information sufficient enough to describe the features of the service. Finally, the property created has the record of the date and time of creation of the service.

What can threaten the privacy of individuals in ubiquitous environments are abusive services, e.g. the sending of ads and ads that are not user interest, user location monitoring, collection of information without permission and unauthorized identification.

To prevent any service has access to personal information of people of a particular environment ubiquitous, class *PrivacyPolicy* (Figure 4) for which an individual can specify which services may have or not have access to your information.

The property *identification* of class *PrivacyPolicy* allows to specify only one privacy policy in the model. To inform about which service is the privacy policy, can be created the property

[1] Closed ubiquitous computing environments are that are physically delimited and where the communication and computing are restricted to those limits

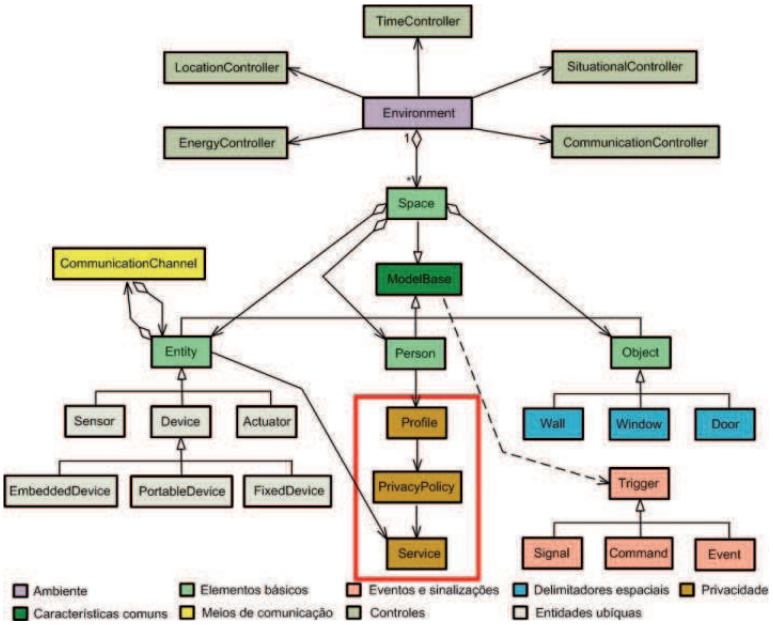


Fig. 2. Main components of the specification

Service

ServiceState

$identification : \mathbb{N}$
$description : TEXT$
$created : TimeStamp$

$\forall x, y : identification \bullet x \neq y$

CreateService

$\Delta(identification, description, created)$
$identification? : \mathbb{N}$
$description? : TEXT$
$created? : TimeStamp$

$identification' = identification?$
$description' = description?$
$created' = created?$

Fig. 3. Characteristics of the class *Service*

service. The *serviceProvider* property allows us to enter a partial set of service providers. For the execution of the service provided by these providers, the *defaultMode* property should be consulted about their modes such as: *allow*, *deny* or *ask*. If a provider is not listed,

PrivacyPolicy

$MODE ::= allow \mid deny \mid ask$

PolicyState

$identification : \mathbb{N}$
$service : Service$
$serviceProvider : \mathbb{P} \downarrow Entity$
$defaultMode : MODE$
$unknownMode : MODE$
$temporalConstraint : \text{seq}(TIME \times TIME)$

Fig. 4. Characteristics of the class *PrivacyPolicy*

the model allows to specify which default mode of execution of the service through the property *unknownMode*. If the mode is *allow* all unknown services have access to personal information, if it is *deny* all unknown services will not have access and, finally, if "ask", the user will have to be consulted about the new service provider and so can determine whether or not to share your information. Finally, based on the section 4.2, it is created the property *temporalConstraint*, which determines which periods of time in which the service will be available or not.
So that the model has a more complete specification of the personal information of users and their preferences, the class *Profile*, Figure 5 has been prepared.

Profile

ProfileState

$identification : \mathbb{N}$
$pseudonym : TEXT$
$personalProperties : \mathbb{P}(TEXT \times \mathbb{F}\, TEXT)$
$servicePolicies : \mathbb{P}\, PrivacyPolicy$
$mixingRate : \mathbb{N}$

$\forall x, y : pseudonym \bullet x \neq y$

Fig. 5. Caracteristicas of the class *Profile*

In the class *Profile*, the property *identification* uniquely identifies a user's profile. Using the concept of mixing Beresford (2005), the property *pseudonym* stores the pseudonym used to identify the user's device and allow the mixing. The property *personalProperties* allows to specify personal preferences such as, for instance, in the category of sports, for instance: soccer, volleyball, basketball; and in the category of movies, for example: action, comedy, romance. The property *servicePolicies* relates the User Profile to their privacy policies, that is, the set of services that may or may not access the information in your personal profile. Finally, *mixingRate* defines a minimal rate of individuals which must be present to occur a mixing. If the rate is zero, this means the user does not want to participate in the mixing. Moreover, to integrate aspects of privacy to the modeling in Campiolo (2005) has been created the property

profile in the class *Person* which allows to assign the class *Profile* to a person; likewise the property *offeredServices* in the class *Entity*, which allows inform which services are provided by a particular entity.

6. The simulation of an environment

This section presents the application of the metamodel presented in section 5 for the scenery of a shopping center. The choice for this application is due to the fact that a shopping center is a sufficiently complex scenery, because it is composed of several other open and closed sceneries (for instance shops, exhibition areas, salons, escalators) and these are well defined from the physical structure shopping.
The goal is to highlight the importance and applicability of metamodel drafted, as well as discuss the various problems leading in this scenario, in the sense to propose some viable solutions. Only the main classes are presented in this section. In addition, in the end of the section is presented a scenery of the shopping center modeled and analyzed in a simulation tool.

6.1 The scenario

As previously presented in Campiolo (2005), the shopping center (Figure 6) corresponds to a scenario consisting of a large amount of people with their devices and several closed and open spaces, clearly delimited physically by the structure of shopping center. There are sensors and distributed devices, monitoring and providing services to individuals within the limits of the internal environment. The services are intended to conquer and provide convenience to users. These services must not be intrusive, i.e. not transgress, the environment because it has an infrastructure to protect the privacy of its users.

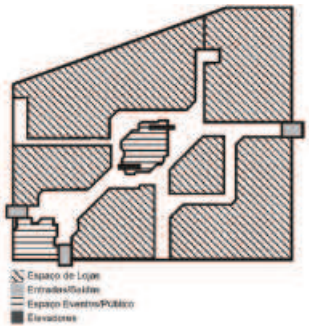


Fig. 6. Illustration of the shopping scene. Source: Campiolo (2005).

The devices and sensors can be located in the stores. They can communicate internally in the store, that is, a client device is detected within the limits of the store, or to communicate a certain distance outside the store. The same principle is valid for the sensors. In addition, sensors and devices can be property of the shopping and can be distributed in other points, being shared by several stores.

6.2 Specific problem

The following scenario illustrates a specific privacy problem in the environment studied. Alice has a profile that is composed of two parts: a set of propositions provided by Alice (profile

A) and a set of propositions inferred by the system or by other entities (Profile B). Alice can determine how to the profile A can be used by the system, in whole or for a particular purpose. On the profile B, his control is very limited, because Alice can not know of its existence. For example, consider that Alice prepares the profile indicating that likes of action movies, gymnastics and self-help books, as shown in Figure 7. The structure named **alice** is the specification of the class *Person* that instantiates Alice in the environment. In this case, the identification **alice** was used as the value of the property *pseudonym*, because Alice had opted out of mixing (property *mixingRate* is zero), i.e., she does not want to maintain your anonymity. This issue will be again discussed in the section 6.5.

Profile : *profile_A*

$identification = 14$
$pseudonym = alice$
$personalProperties = \{sports, \{gymnastics\}\},$
$\{films, \{action\_movies\}\}, \{books, \{self\_help\}\}$
$servicePolicies = marketing\_policy$
$mixingRate = 0$

Fig. 7. Profile A provided by Alice

The Figure 8 presents the privacy policy of Alice, where she determines that your informations are used by advertising service (**marketing_service**), should not be passed to third parties, only to the L and M libraries. In addition, through the property *temporalConstraint*, Alice has defined a temporal restriction alleging that want to be addressed by these services only in time from 8h to 18h.

PrivacyPolicy : *marketing_policy*

$identification = 16$
$service = marketing\_service$
$serviceProvider = sen\_bookstore\_L, sen\_bookstore\_M$
$defaultMode = allow$
$unknownMode = deny$
$temporalConstraint : (8h00m00s, 18h00m00s)$

Fig. 8. Privacy policy for the advertising service

Now, suppose that Alice perform some purchases of books about travel in the bookstore L, one or more times per month for a period of six months. It can be assumed that these books are for herself, it is difficult to be present (unless Alice knows several friends who like to travel and make birthday in this period). Thus, this bookstore is able to create a second profile for Alice, the profile of Figure 9, which is a copy of the profile provided by her, with an additional value in the property *personalProperties*, saying that she likes travel books, fact which she did not reveal personally.
It is important to consider that this information is produced without the knowledge of Alice, it does not fit in the constraint, it will not be provided to third parties and it is a profile more accurate than the profile provided by herself.

Profile : *profile_B*

$identification = 15$
$pseudonym = alice$
$personalProperties = \{sports, \{gymnastics\}\},$
$\{films, \{action\_movies\}\}, \{books, \{self\_help, traveling\}\}$
$servicePolicies = marketing\_policy$
$mixingRate = 0$

Fig. 9. Profile B inferred by system

Now, suppose Alice goes shopping porting a handheld with your unambiguous identification and your profile A stored. At the entrance, Alice is identified, the profile A is read and profile B is rescued by the system. Passing next to the bookstore M, Alice receives a message from some promotion of travel books. This can be approached from various points of view. Alice did not put the information that she likes of travel books in the profile A, because personally prefer to search when she has a specific need. Suppose she travels a lot for work, but can not determine her destination previously. In this case, it is not useful to receive promotion notices. Alice knows that the information placed on your profile A are used for different companies can make customized offers. On the other hand, these offers may be interesting to Alice, they may be more in line with their interests, and provide a good economy in some cases. In any case, the goal of the companies is to sell, which does not necessarily satisfy completely Alice, neither respect your wishes.

6.3 Model and architecture of the environment

The physical structure of the environment remains the same as shown in Figure 6. In this case, are added only new sensors and devices in some areas of the environment. In the inputs and outputs of the environment, are the sensors responsible for collecting the privacy policies of the user. These sensors are called I/O sensors. Only the I/O sensors collect privacy policies. Therefore, it is required an entity to store these policies. The entity responsible for storing security policies is the central server. This server receives from the I/O sensors the information gathered from the user's device. The sensors and local devices access the privacy policies of a client through a connection with the central server.

Thus, the communication protocols with the devices become less stressful and saves the battery from the client device. The same effect is achieved with the collection of the privacy policy file at the entrance. The collection is performed by wireless communication (radio). The distance between the collecting device and client device is small. Therefore, the energy expended and packet loss are much smaller. The I/O sensors must be isolated and have a range of extension, i.e. they must not allow an external sensor retrieve or disrupt privacy policies collection and should ensure that the client device to remain in the range of transmission until the end of the protocol. The last architecture element added are the mixing zones, where none of the users can be located by services. In the environment studied, mixing zones are in the central region and in the exposure environments of the shopping center because, in this places, there is a constant movement of people.

6.4 Understanding the protocols

To facilitate comprehension, the implications and operation of the protocols, these are described based on possible situations that a client is entering into an closed environment of ubiquitous computing.

6.4.1 Entry of a client/user

A client, upon entering the environment carrying his ubiquitous computing device initiates communication with the I/O sensors. The privacy policy file and the client identification are transmitted. The server generates a pseudonym for client registration, which serves as a key to recovery policies. The client device receives this unique identifier and stores it in the corresponding field. The communication is closed.

6.4.2 Detection of a client by a sensor

Upon detecting the presence of a device, the sensors of the environment or establishment, through a simple protocol, gets its pseudonym. After this, consult the central server to determine the privacy policies of user's device. If it is registered the interest of an user for any service provided, a protocol starts to assist him.

6.4.3 Client terminates the communication with a device

A customer, when leaving an establishment or range of a sensor or device, ends links and a mixing mark is redefined. Therefore, while trasiting through a mixing zone, this device will receive a new pseudonym and its mark of mixing again will be redefined.

6.5 Applying the model to a problem

In this section, the problem of Alice (section 6.2) is resumed and the situations described in the problem are applied to the proposed model. The generation of profile B, where is recorded the interest of Alice by travel books, it is inevitable, once that to purchase, Alice is identified, either through credit or otherwise, as the memory of seller or even facial recognition systems used in security cameras in the store.

Given the inevitability of the generation of a profile B, the solution to maintain the privacy of Alice is dissociate her from your profile B. This can be done by assigning a pseudonym to Alice that will be used to identify her. Thus, when passing by the sensors of the store, Alice will not be identified and, therefore, not linked to your profile B. Starting from the idea that Alice wants to have anonymity, the new profile A of Alice with his new pseudonym automatically generated by the central server is shown in Figure 10.

Profile : *profile_A*

$identification = 14$
$pseudonym = 4fasd452$
$personalProperties = \{sports, \{gymnastics\}\}, \{films, \{action\_movies\}\}, \{books, \{self\_help\}\}$
$servicePolicies = marketing\_policy$
$mixingRate = 1$

Fig. 10. Profile A provided by Alice

However, there are situations in which Alice should be identified, for example, to make a payment by credit card. At this moment, the systems can locate Alice, because she is related to the alias pseudonym. The solution is to change your pseudonym again. Forcing a user to leave and re-enter the environment to change the pseudonym is not something plausible. To do this, mixing zones exist, in which Alice can literally mix the crowd, emerging from this zone with a new pseudonym untraceable by commercial establishments. Thus, even in cases where Alice's identity can be inferred by their habits (habits like visiting specific shops in a certain order in a given time), just that Alice passes through a mixing zone to stay anonymous. In the case of shopping center, mixing zones are central points that the user always pass, whether to change level, or to see other stores. Thus, the chances of identification and tracking become very low, except in circumstances where mixing rates are too low.

6.6 The OPNET network simulator

The OPNET (*Optimized Network Engineering Tool*) *Modeler*[2] allows to design and study communication networks, devices, protocols and applications. The models in OPNET are hierarchical. At the lowest level, the behavior of an algorithm or protocol is encoded by a finite-state diagram with embedded code based on C/C++ language. At the intermediate level, discrete functions such as processing, transmission and reception of data packets are executed by separate objects, which behave as defined in a process model. These objects, called modules, are connected to form the network model that, in the hierarchy, is the highest level model. This model, finally, is what defines the scope of a simulation.

6.7 The environment represented in OPNET

For the simulation was considered a simplified environment of the shopping center described in section 6.1. In the simulated environment there are two bookstores, represented by the sensors **Bookstore L Sensor** and **Bookstore M Sensor**; 3 sceneries: the first with 50 users, the second with 100 users and third with 300 users which are represented by their mobile devices; the central server or **Server**; a mixing zone represented by **Mix Sensor** and a I/O sensor that is **IO Sensor**. This elements are presented in the OPNET network model, in the Figure 11 .

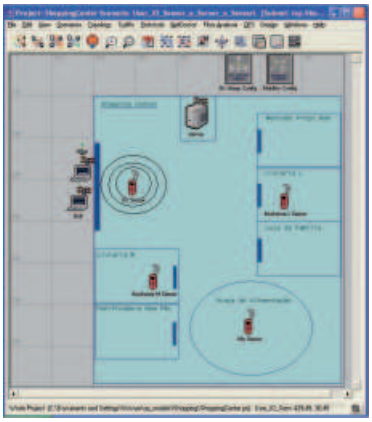


Fig. 11. Shopping center scenario in Opnet.

[2] www.opnet.com

The following describes the exchange of messages as well the element modules of the shopping center.

6.8 Packages for exchange of messages

For the exchange of messages between the elements of the shopping center were developed the following packages:

- shopping_sensor_requests_pck or request package;
- shopping_profile_pck or profile package;
- shopping_pseudonym_pck or pseudonym package;
- shopping_new_pseudonym_pck or new pseudonym package, created by mixing sensor and useful only for users who wish to participate in the mixing;
- shopping_service_policies_of_user_pck or service policies of user package;
- shopping_mixing_pck or mixing package;
- shopping_marketing_pck or marketing package.

6.9 Model of I/O sensor node

Each network node represented in OPNET consists of a *node-model* and a *process-model*. The model of the I/O sensor node is shown in Figure 12. It is possible perceive that there are two input flows (**stream 0** and **stream 1**) and one output flow.

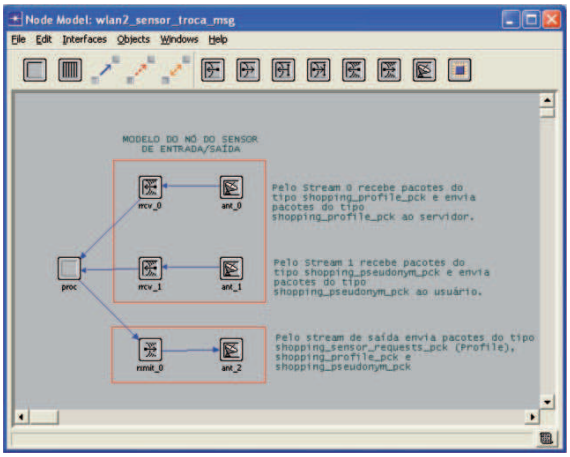


Fig. 12. Model of I/O sensor-node

The sensor-node is responsible for capturing user information and forward it to the server, and vice-versa. The sensor-node periodically sends request packets or **shopping_ sensor_requests_ pck** of type *Profile* to the new shopping users to start exchanging messages. If a user responds to these requests, it is by the flow 0 that the sensor will receive the packets of type **shopping_profile_pck** from the user and will send by the output flow to the server. The server then generates a pseudonym for this user and send it to the I/O sensor. It is by flow 1 that the sensor will receive the packages of type **shopping_pseudonym_pck** from the server and it sends by the output flow to the user.

The Figure 13 presents the process-model of the I/O sensor. There are four states: **start**, **send**, **wait** e **end**. The state **start** loads a structure variable called **Address**, with IP values informed in the node interface. The I/O sensor node, as all other environmental sensors, works with two BSS: 0 and 1. The BSS 1 is for communication with the user and the BSS 0 is for communication between the sensors and the server.

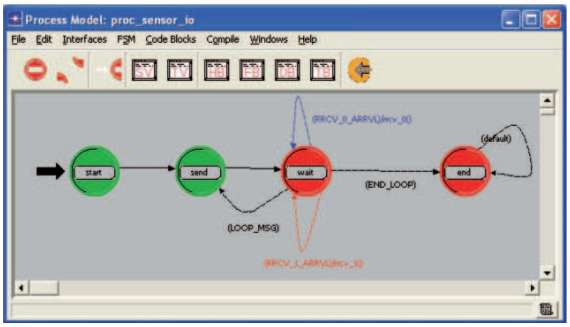


Fig. 13. Process model of I/O sensor.

The state **send** is what sends request packets to new users of the shopping, which are packages **shopping_sensor_requests_pck** of type **Profile**. After send this request, the state machine assumes the State **wait**. In this state the sensor waits packages from the user (flow 0) or from the server (flow 1). This state can also return to the state **send** to send new requests (transition **LOOP_MSG**) or go to the final state **end** with the transition model.

The user-node model is the most important, because it represents the key element of this work, which is the user and how to maintain your privacy. As shown in Figure 14, the user's device has six input flows (**stream 0**, **stream 1**, **stream 2**, **stream 3**, **stream 4** e **stream 5**) and a output flow.

By the flows 0, 1 and 4, the user receives request packets, which are packets of type **shopping_sensor_requests_pck** of the I/O sensor, bookstore sensor and mixing sensor, respectively. After receiving request packets from the I/O sensor, the user sends the packet with your profile **shopping_profile_pck** to the sensor, so that the user can receive a pseudonym. As explained in section 6.9, the sensor sends this packet to the server, which generates a pseudonym for the user, and that is sent by the I/O sensor. The pseudonym is received by user through the flow 2. Upon receiving the pseudonym the user's device writes in its internal memory the value that can be used in other communications during their stay at the shopping.

After receive the pseudonym, the user is able to answer the requests of the bookstores sensors, which are the requests made by flow 1. Then, the bookstore sensor will query the server to check the permissions to send advertisements to the user and to find out the user preferences. If it is possible to send out advertisements, the user will receive by the flow 3, and if there is any promotion that is of interest, certainly, he will go to the bookstore.

If the user walks near an mixing zone, they receive, by flow 4, a request about your mixing rate for the mixing zone sensor determine if the user wants or not join the mixing. Your mixing rate value is sent in the **shopping_mixing_pck** packet. If the value of mix rate is 1, the sensor will request the server a new pseudonym that is sent to user. This new pseudonym is received by flow 5 in the packet **shopping_new_pseudonym_pck**. This causes any parallel profile generated by the stores is lost and only the user-generated profile is respected, maintaining so

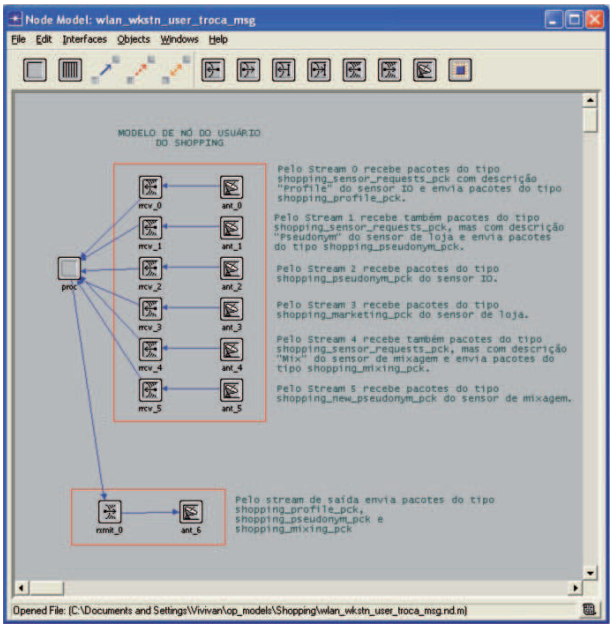


Fig. 14. Users node model.

your privacy.
In addition to the user-node model, there is also a process-model, shown in Figure 15 . There are two states: **start** e **wait**. The state **start** loads internally the user's IP adress and the state **wait** is responsible for waiting the communications in all flows described above.

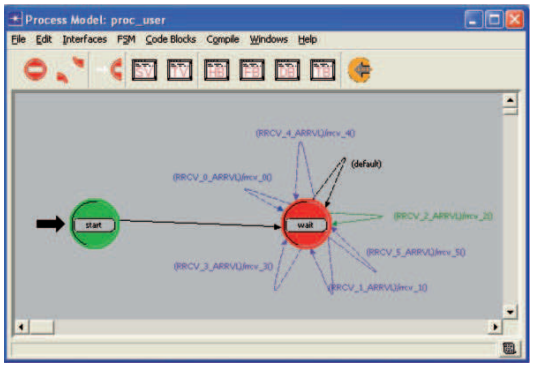


Fig. 15. User node model.

6.10 Server-node model

The server-node model is responsible for storing the user's profile and generate a pseudonym for this user, as soon as he enters in the shopping. The server-node model has three input flows and a output flow.

By flow 0, the server receives the user's profile from the I/O sensor through the packets **shopping_profile_pck** and it stores in your user list. After store the user, the server generates a pseudonym, store it in the user registration and sends it to the user by the I/O sensor in the packet **shopping_pseudonym_pck**. The flow 1 is for communications between the sensors of bookstores with the central server. The bookstore sensor sends to the server the pseudonym of the user, to locate your personal preferences, privacy policies are used and checked if the sensor has permission to send an ads to the user. If it has permission, the server sends the personal preferences and user privacy policies in the packet **shopping_service_policies_of_user_pck**. The flow 2 is for the server receive requests from mixing sensor to generate new pseudonyms for the shopping clients who wish to participate in the mixing. In this case, the incoming packet is the **shopping_sensor_ requests_pck** of type **New Pseudonym**, and the server sends the new pseudonym to mixing sensor in packet **shopping_new_pseudonym_pck**.
As in the user-process, the server process has two states: **start** and **wait**. In **start** state, the server's IP variable is loaded and int the state **wait**, the server waits for packets to establish a communication with the sensors.

6.11 Sensor-node model of bookstores

The bookstore sensor-node is interested in obtaining the user's personal information to send advertising services. Its sensor-node model has two input flows and one output flow.
The sensor-node periodically sends pseudonym request messages, which are the **shopping_sensor_requests_pck** packets, to the users who are nearby. The response to this request is received in the flow 0, in the packet **shopping_pseudonym_pck**. This pseudonym is sent to the server to the sensor obtain the preferences and service policies provided by the user. So this informations are received by the flow 1, in the packet **shopping_service_policies_of_user_pck**. At receive this packet, the sensor verifies the advertisements that can interest the user and sends in the packet **shopping_marketing_pck**.
The process-model of the bookstore sensor-node is similar to I/O sensor and has four states: **start**, **send**, **wait** and **end**. The state **start** loads the sensor's IP adress internally. The state **send** sends request packet of the type **shopping_sensor_requests_pck**, and wait the response of this requests in the state **wait**. Finally, the **end** state finishes the processing.

6.12 Mixing zone

The mixing zone is represented in the environment by the mixing sensor. It checks which users are interested in change his pseudonym, avoiding that he is associated with any secondary profile created by stores, as explained in section 6.5.
The mixing sensor periodically sends **shopping_sensor_requests_pck** packets of type **Mix** to the near users, to response with its mixing rate. The packet **shopping_mixing_pck** is received by the input flow and processed. If the mixing rate is 1, the mixing sensor sends the user pseudonym to the server, to create a new pseudonym. This request is made sending the packet **shopping_sensor_requests_pck** of type **New Pseudonym** to the server. After receive the new pseudonym generated by the server in flow 1, the sensor in packet **shopping_new_pseudonym_pck**, sends to the user, who wants to join the mix.

6.13 Anonymity measures

Anonymity is the state of being not identifiable within a set of subjects, that is: the anonymity setPfitzmann & Köhntopp (2001). For the shopping example, the definition given by

Nussbaumer (2007) clarifies what is the set of anonymity: the group of people visiting a mixing zone during the same period. The higher the number, the greater the degree of anonymity offered. When the anonymity set is reduced to one element, the user is fully exposed and loses all its anonymity. However, the user can deny the information of his location to an application until the mixing zone offers a minimum level to anonymity. This procedure was not implemented in the shopping scenario. According to Toth et al. (2004), the first studies were aimed to quantify the anonymity level provided, as the size of anonymity set Berthold et al. (2000). However, this is not a good measure of anonymity, by considering that the probabilities might not be uniformly distributed.
With the need to measure the degree of anonymity, Serjantov and Danezis Serjantov & Danezis (2002) introduced the entropy degree as a measure of anonymity. The following model was presented by them:
Definition 1: Given an attack-model and a finite set of all users Ψ, be $r \in R$ a function for an user (R = sender, address) with respect to a message M. Let U the probability of the user $u \in \Psi$ being attacked having a function r with respect to M. With this definition, the measure of anonymity of the sender and of the receiver can be defined as:
Definition 2: The size S of a probability distribution U of the anonymity r is equal to the entropy of the distribution. In other words:

$$\boxed{S = -\sum_{u=1}^{\Psi} p_u \log_2 p_u} \quad (1)$$

where $p_u = U(u, r)$.
This type of entropy is known as simple entropy Toth et al. (2004). In Diaz et al. (2002) was followed a different approach, where only the anonymity of the sender is considered. A represents the set of anonymity of a certain message M, i.e. $A = \{ u \mid (u \in \Psi) \wedge (p_u > 0) \}$. In addition, let N the anonymity set size, i.e. $N = |A|$.
Definition 3: The anonymity degree provided by the system is defined by:

$$\boxed{d = \frac{H(X)}{H_M}} \quad (2)$$

Where $H(X) = S$ e $H_M = \log_2 N$. For a particular case with one user, d is assumed to be zero. This measure is known as normalized entropy. In both cases, zero means no anonymity, ie, the attacker knows 100% the sender of the message.
In the simple case of entropy, the maximum anonymity is achieved when $S = \log_2 N$ and in normalized when $d = 1$ Toth et al. (2004). In this chapter will be used the normalized entropy metric, since the major interest is in the anonymity of the sender, i.e. of the shopping user and not the other elements.

6.14 Scenarios for simulation

To obtain the maximum degree of anonymity of the shopping were considered 3 scenarios: one with 50 clients, one with 100 customers and, finally, a scenario with 300 customers. This represents the number of users that may be present in the mixing zone, but not necessarily want to join the mix. For each scenario, some situations were simulated, by varying the number of users who wish to join the mix, as shown in Table 1.
The attack-model for these scenarios is similar to that presented by Diaz et al. (2002), for the *Onion Routing* case. In this model, N is the size of the anonymity set, and the maximum entropy for this N users is:

Users in mix zone	Users participating in the mix
50	1 2 10 25 40 50
100	1 2 20 50 80 100
300	1 2 60 150 240 300

Table 1. Situations for simulation.

$$H_M = log_2 N \tag{3}$$

In the case of shopping, N is the number of users that are in the mixing zone of shopping, not even having the desire to participate in the mixing zone. In this case, the attack is characterized as a sensor of the stores trying to associate the identity of a client with a secondary profile for this. The set that contains only users interested in participating in the mix is called set A, where $1 \leq A \leq N$. In this case, the probability distributions for the users A are uniform:

$$p_i = \frac{1}{A}, 1 \leq i \leq A; p_i = 0, A + 1 \leq i \leq N \tag{4}$$

So that, the entropy and the degree of anonymity are defined as:

$$H(X) = log_2 A, d = \frac{H(X)}{H_M} = \frac{log_2 A}{log_2 N} \tag{5}$$

To apply this attack-model to the presented scenarios, we are considering various sizes for the set A, as previously shown in Table 1. The degree of anonymity obtained for each scenario is presented in Table 2.
The results obtained with the simulations of the three scenarios were summarized and compared in Figure 16. In all situations where there is only one client, there is no guaranteed anonymity for the client. Even with two users, the degree of anonymity achieved is very low. In Diaz et al. (2002) is suggested an intuitive value to the minimum degree of anonymity for a system to provide adequate anonymity. This value is d $\geq$ 0,8. The situations for the samples of sets A being approached, which gives a degree of anonymity $\geq$ 0,8 are the situations where A $\geq$ 25, for the scenario where N = 50 users, A $\geq$ 25, for N = 100 users and A $\geq$ 150, where there are 300 users. In addition, the maximum degree of anonymity is obtained when N = A.

Set N	Set A	p_i	$log_2 N$	$log_2 A$	d
50	1	1,0000	5,6439	0,0000	0,0000
	2	0,5000	5,6439	1,0000	0,1772
	10	0,1000	5,6439	3,3219	0,5886
	25	0,0400	5,6439	4,6439	0,8228
	40	0,0250	5,6439	5,3219	0,9430
	50	0,0200	5,6439	5,6439	1,0000
100	1	1,0000	6,6439	0,0000	0,0000
	2	0,5000	6,6439	1,0000	0,1505
	20	0,0500	6,6439	4,3219	0,6505
	50	0,0200	6,6439	5,6439	0,8495
	80	0,0125	6,6439	6,3219	0,9515
	100	0,0100	6,6439	6,6439	1,0000
300	1	1,0000	8,2288	0,0000	0,0000
	2	0,5000	8,2288	1,0000	0,1215
	60	0,0167	8,2288	5,9069	0,7178
	150	0,0067	8,2288	7,2288	0,8785
	240	0,0042	8,2288	7,9069	0,9609
	300	0,0033	8,2288	8,2288	1,0000

Table 2. Degree of anonymity obtained for the scenarios

Based on the results presented in Figure 16 , can say that the degree of anonymity of an environment increases as the number of elements in the set A increases. That is, the more users want to participate in the mix, harder for an attacker to discover the identity of a user.

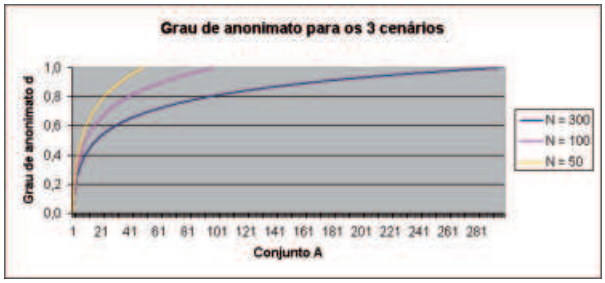


Fig. 16. Comparison between the 3 scenarios.

A problem found in metric of calculation of the degree of anonymity is that, when considering an environment where $N = 2$ and $A = 2$, there is $d = 1$. That means that we obtained the maximum degree of anonymity for that environment, considering $p_i = 0,5$. However, in practice, it is known that $N = 2$ does not guarantee the anonymity of these two users, even with $A = 2$. For this reason, it is important always consider higher values for N and A. An alternative could be to set a minimum value for the size of A, and the user only accept participate of the mix when the minimum number of users for the set A is reached.

7. Conclusion

The model developed for dealing with privacy issues in closed ubiquitous computing environments presents a solution, based on service constraints through user-defined privacy

policies. This is satisfactory to solve the problems of privacy invasion caused by services offered in such environments.
The model of architecture and data developed satisfies the requirements, avoiding unnecessary packet traffic and wasting battery of client devices.
The discussion of privacy issues led to a reflection on which future concerns and precautions that users and applications should consider for the use of devices and ubiquitous computing environments.

8. References

Aoyama, T. (2008). A new generation network - beyond ngn, *Innovations in NGN: Future Network and Services. First ITU-T Kaleidoscope Academic Conference* pp. 3–10.

Beresford, A. R. (2005). Location privacy in ubiquitous computing, *Technical Report UCAM-CL-TR-612*, University of Cambridge, Computer Laboratory.

Beresford, A. R. & Stajano, F. (2004). Mix zones: User privacy in location-aware services, *Pervasive Computing and Communications Workshops, IEEE International Conference on* 0: 127–131.

Berthold, O., Federrath, H. & Kpsell, S. (2000). Web mixes: A system for anonymous and unobservable internet access, *Designing Privacy Enhancing Technologies*, Springer-Verlag, pp. 115–129.

Bhaskar, P. & Ahamed, S. I. (2007). Privacy in pervasive computing and open issues, *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*, IEEE Computer Society, Washington, DC, USA, pp. 147–154.

Campiolo, R. (2005). *Aspectos de modelagem de ambientes de computação ubíqua*, Master's thesis, Universidade Federal de Santa Catarina.

Campiolo, R., Cremer, V. & Sobral, J. B. M. (2007). On modelling for pervasive computing environments, *in Proceedings of 10th International Symposium on Modelling, Analyses and Simulation of Wireless and Mobile Systems - MSWiM 2007* pp. 3–10.

Cheng, H. S., Zhang, D. & Tan, J. G. (2005). Protection of privacy in pervasive computing environments, *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, Vol. 2, IEEE Computer Society, Washington, DC, USA, pp. 242–247.

Diaz, C., Seys, S., Claessens, J. & Preneel, B. (2002). Towards measuring anonymity, *in* R. Dingledine & P. Syverson (eds), *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, Springer-Verlag, LNCS 2482, pp. 54–68.

Duke, R., King, P., Rose, G. & Smith, G. (1991). The object-z specification language: Verson 1, *Technical Report 91-1*, University of Queensland.

Jiang, X. & Landay, J. A. (2002). Modeling privacy control in context-aware systems, *IEEE Pervasive Computing* 1(3): 59–63.

Langheinrich, M. (2001). Privacy by design — principles of privacy-aware ubiquitous systems, *j-LECT-NOTES-COMP-SCI* 2201: 273–??

Lederer, S., Dey, A. K. & Mankoff, J. (2002). A conceptual model and a metaphor of everyday privacy in ubiquitous computing environments, *Technical Report UCB/CSD-02-1188*, EECS Department, University of California, Berkeley.
URL: *http://www.eecs.berkeley.edu/Pubs/TechRpts/2002/5464.html*

Myles, G., Friday, A. & Davies, N. (2003). Preserving Privacy in Environments with Location-Based Applications, *IEEE Pervasive Computing* 2(1): 56–64.

Nussbaumer, M. (2007). Location privacy.
URL: *www.sec.informatik.tu-darmstadt.de/pages/lehre/SS07/sem_misc/papers/nussbaumer.pdf*

Pfitzmann, A. & Köhntopp, M. (2001). Anonymity, unobservability, and pseudonymity - a proposal for terminology, *Designing Privacy Enhancing Technologies*, Springer-Verlag, pp. 1–9.
URL: *http://dx.doi.org/10.1007/3-540-44702-4_1*

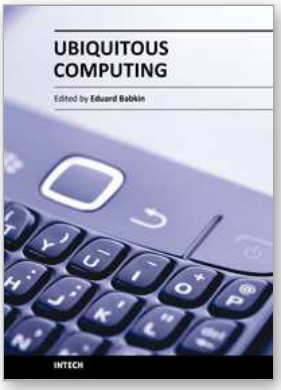
Serjantov, A. & Danezis, G. (2002). Towards an information theoretic metric for anonymity, *Proceedings of Privacy Enhancing Technologies (PET2002)*, Springer-Verlag, pp. 41–53.

Spivey, J. M. (1989). *The Z Notation : A Reference Manual*, Prentice Hall.

Toth, G., Hornak, Z. & Vajda, F. (2004). Measuring anonymity revisited, *in* S. Liimatainen & T. Virtanen (eds), *Proceedings of the Ninth Nordic Workshop on Secure IT Systems*, Espoo, Finland, pp. 85–90.

Weiser, M. (1993). Some computer science issues in ubiquitous computing, *Communications of the ACM* 36(7): 75–84.

Weiser, M., Gold, R. & Brown, J. S. (1999). The origins of ubiquitous computing research at parc in the late 1980s, *IBM Systems Journal* 38(4): 693–696.



Ubiquitous Computing

Edited by Prof. Eduard Babkin

ISBN 978-953-307-409-2
Hard cover, 248 pages
Publisher InTech
Published online 10, February, 2011
Published in print edition February, 2011

The aim of this book is to give a treatment of the actively developed domain of Ubiquitous computing. Originally proposed by Mark D. Weiser, the concept of Ubiquitous computing enables a real-time global sensing, context-aware informational retrieval, multi-modal interaction with the user and enhanced visualization capabilities. In effect, Ubiquitous computing environments give extremely new and futuristic abilities to look at and interact with our habitat at any time and from anywhere. In that domain, researchers are confronted with many foundational, technological and engineering issues which were not known before. Detailed cross-disciplinary coverage of these issues is really needed today for further progress and widening of application range. This book collects twelve original works of researchers from eleven countries, which are clustered into four sections: Foundations, Security and Privacy, Integration and Middleware, Practical Applications.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vivian C. Kalempa, Rodrigo Campiolo, Lucas Guardalben, Urian K. Bardemaker, João Bosco M. Sobral (2011). On Modeling of Ubiquitous Computing Environments Featuring Privacy, Ubiquitous Computing, Prof. Eduard Babkin (Ed.), ISBN: 978-953-307-409-2, InTech, Available from: http://www.intechopen.com/books/ubiquitous-computing/on-modeling-of-ubiquitous-computing-environments-featuring-privacy

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821