

An ubiquitous and non intrusive system for pervasive advertising using NFC and geolocation technologies and air hand gestures

Francisco M. Borrego-Jaraba*, Irene Luque Ruiz and Miguel Ángel Gómez-Nieto
Department of Computing Science and Numerical Analysis, University of Córdoba, Córdoba, Spain

Abstract. In this paper we present a pervasive proposal for advertising using mobile phones, Near Field Communication, geolocation and air hand gestures. Advertising post built by users in public/private spaces can store multiple ads containing any kind of textual, graphic or multimedia information. Ads are automatically shows in the mobile phone of the users using a notification based process considering relative user location between the posts and the user preferences. Moreover, ads can be stored and retrieved from the post using hand gestures and Near Field Communication technology. Secure management of information about users, posts, and notifications and the use of instant messaging enable the development of systems to extend the current advertising strategies based on Web, large displays or digital signage.

Keywords: Advertising, air hand gesture, geolocation, near field communication, pervasive applications

1. Introduction

The concept of “Smart Cities” has attracted considerable attention in the context of urban development policies. The Internet and broadband network technologies as enablers of e-services are becoming more and more important for urban development while cities are increasingly assuming a critical role as drivers of innovation in areas such as health, inclusion, environment and business [1].

The evolution towards a future information and knowledge society is characterized by the development of personalized individual as well as collective services that exploit new qualities of infrastructures and components situated in a smart environment [2]. They are based on a range of ubiquitous and pervasive communication networks providing ambient computing and communication at multiple levels.

Ambient intelligence (AmI) represents a vision of the future where intelligent or smart environments and systems react in an attentive, adaptive and active way to the presence and activities of humans and objects in order to provide intelligent/smart services to the inhabitants of these environments. AmI paradigm aims to develop environments able to interact with the user in an autonomous way in order to make life easier for people in different fields [3]. Olaru et al. [4] see an Ambient Intelligence environment

*Corresponding author: Francisco M. Borrego-Jaraba, Department of Computing Science and Numerical Analysis, University of Córdoba, E-14071 Córdoba, Spain. E-mail: fborrego@uco.es.

as a large number of devices that serve the needs of their respective users. The devices are mostly going to deal with information: delivering relevant information to interested users, aggregating, filtering and reasoning about the information.

Derived from the concept of Ubiquitous Computing introduced by Weiser [5], the term of Ambient Intelligence was coined at the beginning of the 21st century to describe a ubiquitous electronic environment that would pro-actively, but sensibly and non-intrusively support people in their daily lives [6].

A technology that facilitates the achievement of the AmI objectives is Near Field Communication (NFC) [7]. NFC simplifies the users interactions with the context-aware-services offered in intelligent environments, also promoting a new interaction model, called “*touch paradigm*”.

It is difficult to talk about Ambient Intelligence without mentioning context-awareness. Context-awareness means that one is able to use context information. A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context of use [8]. The term context-aware computing is commonly understood by those working in context-aware, where it is felt that context is a key in their efforts to disperse and transparently weave computer technology into our lives. One goal of context-aware systems is to acquire and utilize information in the context of a device in order to provide services that are appropriate to the particular people, place, time, event, etc. These systems aim to provide context-aware access to information, communication and computation.

The most critical aspects of context-aware are location. Location-awareness concerns the use of information about an individual’s current location to provide more relevant information and services to that individual [9]. Location-awareness is a special type of context-awareness. Location-aware computing environments offer the capability for automatic, regular, and real-time sensing of a person’s location with a high degree of spatial and temporal precision and accuracy. Together with technological advances in mobile computing and wireless communication, which enable rapid processing and communication of location information, these developments allow the location of mobile individuals to be tracked in a way never before possible.

This paper presents a framework called WingPosts [10]. WingPosts is a ubiquitous system which enables users to spread their ads at any time, any place and directed to the appropriate user. Thus, the visibility of your products and services is increased instantaneously. WingPosts integrates NFC technology, geolocalization and hand gestures. With WingPosts, the ads will be located in virtual posts which allow users to access ads anytime and anywhere.

Virtual posts built by users in public/private spaces can store multiple ads containing any kind of textual, graphic or multimedia information. Ads are automatically shown on the mobile phone of the users using a notification based process considering relative user location between the posts and user preferences.

The paper begins with a review of the research on digital advertising. It is followed by a description of the context. In Sections 4, 5 and 6 we present the data model, the interaction model and the pervasive functional model respectively. Section 7 describes the system test. WingPosts system (Website and Mobile) is described at the Section 8. Finally, Section 9 presents the conclusions.

2. Related works

Most people have become used to the traditional way of experiencing and consuming advertisements as part of their daily lives. The success of advertisements and their value for the individual is defined by type, content and placement (in terms of location and time) [11].

Mobile advertisements are effective due to user's immediate and personal attention to mobile devices. When combined with location-based information, the effectiveness is shown to be even higher. Many industry surveys estimate the potential market for mobile advertisements in billions of dollars per year. There is a need to design and evaluate artifacts to study mobile advertising and transactions [12].

Omntrakul et al. [13] reports a recent development of one advertising system (called AdTouch) based on QR barcodes. This paper offers a set of comprehensive services for advertisers and publishers to market their printable 2D-barcode ads in different media, including barcode-based ad posting, distributing, detecting and capturing using Android mobile devices.

Aalto et al. [14] introduce a novel B-MAD (Bluetooth Mobile Advertising) system for delivering permission-based location-aware mobile advertisements to mobile phones using Bluetooth positioning and Wireless Application Protocol (WAP) Push. The system was not specifically designed with privacy and security issues in mind. Device address data sent from a Bluetooth sensor to the server is not encrypted.

Haddadi et al. [15] introduce MobiAd, a scalable, location-aware, personalized and private advertising system for mobile platforms. They present a first proposal for personalized, localized and targeted advertising on smart phones. Utilizing the wealth of information available on the phone, MobiAd presents the user with local ads in a privacy preserving manner.

Location-aware mobile advertising involves sending content-rich messages to users in certain locations, where size of the location identified by advertisers or users, or to users in any location, using user preferences. The effectiveness of location-aware mobile advertising is high especially if these messages can be highly targeted and time-specific. The effectiveness of location-aware advertising can be further improved by using contexts derived from multiple sets of information.

The basic ideas behind context-awareness are to create pro-active and smart operation of artifacts by minimizing user efforts and interactions, creating a very high level of intelligence in the systems, adding adaptability and effective decision making, and increasing the level of customization and personalization for users [12].

3. Description of the context

Dey et al. [16] proposes a general definition of context as: "*any information that can be used to characterize the situation of an entity*". An entity is a person, place or object that is considered relevant to the interaction between the user and an application, including the user and the application themselves and, if the application is context-aware, it should provide relevant information and/or services to the user, where relevancy depends on the user tasks.

Context definition implies the enumeration and full descriptions of all actors involved in the application. The depth and characteristics of this definition depend on the actors' nature and the tasks of these actors' development in the systems. Formal definitions of the context have been proposed [17] showing the complexity of this activity for pervasive and context-aware applications. Benou et al. [18] proposes that in order to facilitate the design and development processes of applications a solid perception of concepts and structures related to context capturing and management is required, defining that the context information of an m-commerce application is every piece of information which may be used to characterize a state of an entity, which can be considered to be relevant to the interaction of the user with the particular application. Authors classify the entities involved in the context in four groups: user domain, computing domain, environment domain and application-specific domain, and they propose the use of Entity-Relationship diagrams to represent the relationships between them, and also to classify the

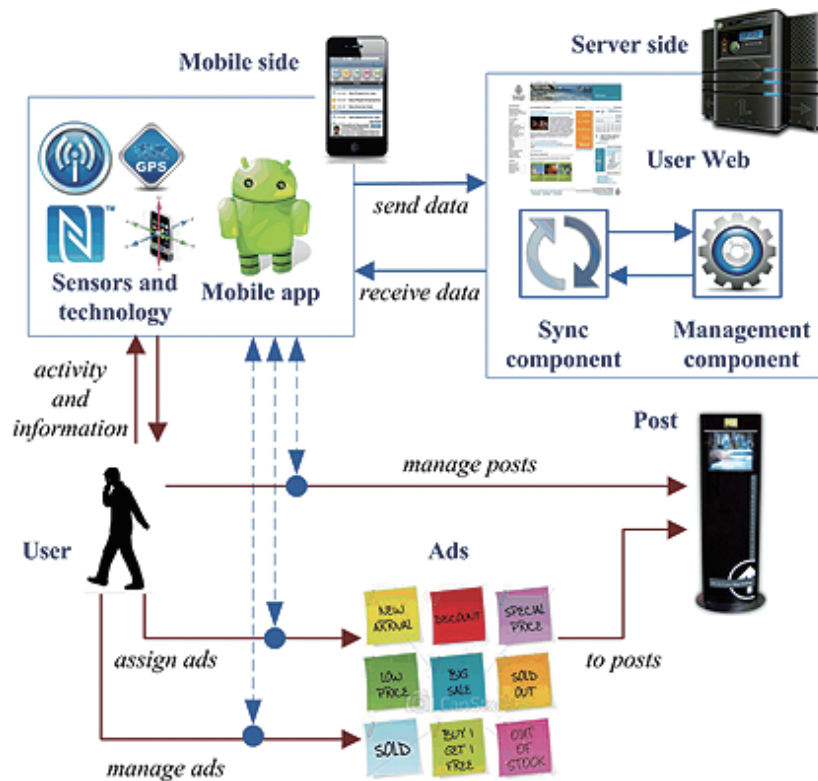


Fig. 1. Context description showing the main entities and their relationships.

relationships in each of these groups, considering the group in which the main entity involved is. This is a rigid classification, so in the pervasive mobile application an entity can be considered in more than one of these domain groups, taking two aspects into consideration: a) the real location of the entity, and b) the virtuality of the entity, that is, if in the behaviour of the entity it acts as real or virtual.

3.1. The main elements of the context

Our proposal considers the following main entities (see Fig. 1): users, posts and ads, and two entities classified within the computing domain: the mobile device and the server side.

A user is any person using the application (here, we do not consider administrator roles). A user publishes ads and/or receives ads from the surrounding environment, that is, the real world. User preferences and user grants determine the behaviour of the user in the environment and how other entities can relate to the user.

The posts are signages used by users to publish ads. Signages are located in the environment and their properties characterize the type of ads that the posts can support and the way of the users can receive information from them.

From the environment point of view, two types of posts are considered: a) real posts, and b) virtual posts. A real post is any physical pole or signage located in the real world (i.e. JCDecaux posts [19]), whereas virtual posts are not physical posts. Any of these types of posts have associated a location in the real world, and a set of properties determining their relationships with users and ads, the actions that the users can perform with the posts, and how these actions can be carried out.

In our proposal, real posts can be located in indoor and outdoor spaces; however, virtual posts can only be located in outdoor spaces. Moreover, the interaction of the users with posts depends on the post type: a virtual post only supports gesture based interactions, while a real post supports gestures and NFC based interactions. On the other hand, posts can be public and private: a public post allows any user to use it to store ads, while a private post has its use restricted to its owner. Virtual and real posts can be either public or private.

Ads are advertising “hanging in the posts” by users with the aims of being disseminated to the largest number of people. Ads can contain any kind of information (textual, graphic, and multimedia) determining the user preferences as to how this information would be received.

3.2. *The computing entities of the context*

Two computing elements have been considered (see Fig. 1): a) the server side, and b) the mobile side. The server side is composed of a set of subsystems or applications aimed at the management of the information (management component), the communication with the mobile side (synchronization component) and the publicity of the information to the users (Web portal).

The management component is responsible for the context management. The information about the context elements is stored and managed by this component. The management component stores information consisting of identification, state, tasks and actions performed with other entities, so the mobile side is only in charge of managing the basic information necessary for the interaction of the user with the environment.

The synchronization component is responsible for the communication between the server and the mobile side. This communication ensures the reliability of the information allowing the interaction of the users with the remaining context entities. Finally, the web portal is devoted to the management of ads and posts information and user registration.

The mobile side is composed of two components: a mobile phone and the mobile application. The mobile phone should be provided with movement sensors, such as an accelerometer and a compass, in order to support air hand gesture based interactions, and NFC chip to supports Near Field Communication interactions. In addition, mobile phones should support GPS and wireless technologies.

Mobile applications running in the mobile phone is in charge of supporting user interaction with the environment and communication with the server side.

3.3. *The relationship between the entities of the context*

The entities of the context are related to each other by means of the user activity. User activity is gathered by the mobile device through the mobile application using different mobile components (movement sensors, GPS and NFC chip) and application interface.

Application interface allows users the management of personal information, and the information related to own posts and ads. The user creates ads and assigns them to public posts and posts created by the user, by means of the mobile application interface. Furthermore, the application allows the user to interact with the user Website.

In addition, the assignation of ads to posts can be performed according to the post characteristics. Indoor posts, always real posts, should support RFID tags devoted to storing tag identification. Thus, NFC technology is used for interacting with these posts: the user touching the post with the phone enabled with NFC can store and retrieve the ads from the post. In the case of outdoor posts (real or virtual posts), the user can interact with them using NFC, only if the post is augmented with a tag, and

Table 1
Relationship between the active entities of the model and description of activities and participating technology

Entity	Activities	Related entity	Participating technology	Description
User	Manage user data	User Web, mobile phone	Web, Wireless	Interact with the user web portal for registration, management of personal data, information and download of the mobile application
	Manage posts	Post, mobile app, sync component	GPS, NFC, Wireless, Movement sensors	Get the user GPS to be used as post location store and retrieve post data and ads communicate with sync component sending and retrieving post data
	Show posts	Post, mobile app, sync component	GPS, NFC, Wireless, Movement sensors	Retrieve posts information based on user location from server side using the sync component
	Manage ads	Ads, mobile app, sync component	NFC, Wireless	Store and retrieve ads data communicate with sync component sending and retrieving post data
	Assign ads	Ads, Posts	NFC, Wireless	Store and retrieve ads data in posts communicate with sync component sending and retrieving information
	Show ads	Post	GPS, Wireless, NFC, Movement sensors	Retrieve ads information from posts based on user location from server side using the sync component
Server side	Manage data	Sync component	Web, Wireless	Manage information in server database and respond to requests sending by the sync component
Sync component	Request dispatcher	Mobile app, server side	Wireless	Receive request from mobile app, transfer the request to the server side, receive the response and send back the answer to the mobile app
User Web	Manage user data	User	Web, Wireless	Manage user data for registration and downloading of the mobile application

by means of air hand gestures. Gesture communication allows the user to interact with the posts without needing to touch it. Thus the user can store and retrieve ads from outdoor posts with two simple gestures: throwing, for storing, and shaking for retrieving ads.

Finally, users and the server side are related by means of the synchronization component. User actions of storing and retrieving are sent by the mobile application to the synchronization component. This component is in charge of initiating the necessary agents of the server side to respond to the user request, receiving the result and sending it back to the mobile application to be shown to the user.

In addition, the server side interacts with the mobile application using the notification system. User preferences determine the latency of this process consisting of sending information about the user location to the synchronization component (gathered from the GPS sensor) and user preferences. This information is used by the server side to retrieve information on ads and posts matching the query and send it back to the user as notifications.

While posts and ads are non-active entities, a set of activities or functionalities should be modelled for the remaining active entities. Table 1 shows the relationship between the entities for some of the main functionalities that the active entities should perform. Each activity requires the use of different

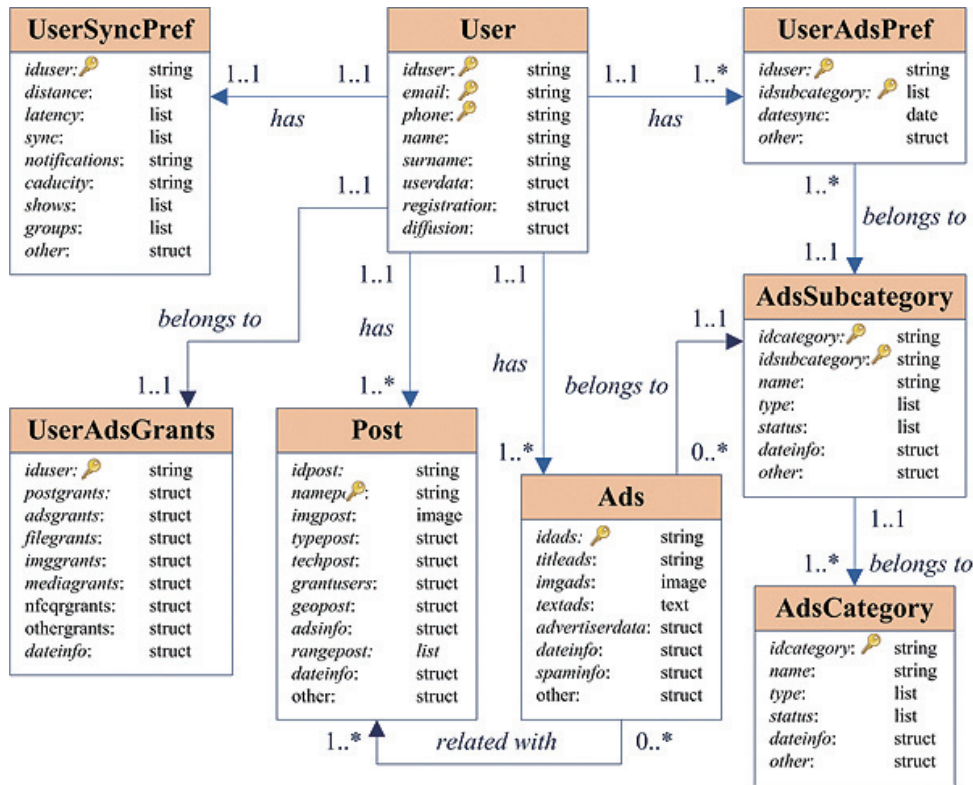


Fig. 2. Main information objects of the system.

technology to be performed. If technology is available in the mobile device, the mobile application should set it without user participation; if the technology is not available in the mobile device or it cannot be used for any reason, then the activity cannot be performed.

4. Data model

The system described in this paper is basically a proposal for pervasive advertising based on the diffusion of personalized ads by means of virtual and real ads signages. The pervasive advertising is carried out using novelty technology such as geolocation, NFC, as well as easy user interfaces based on air hand gestures.

As Fig. 2 shows, the user information stored in the system is closely similar to any other instant messaging or web advertising system. Users are uniquely identified by different attributes defined as unique keys: a) *iduser*, system-defined user identification, b) *email*, and c) *phone*. Thus, no different users can be registered using the same email or phone number, because communication and advertising notifications to users are carried out using these attributes. The *Users* class also involves other attributes devoted to storing information about registration, status, user data, and so on.

Communication between users, environment, mobile application and the server side is performed following the set of user preferences managed by the class *UserSyncPref*. The attribute *distance* stores information on the behaviour of the user in the environment, this attribute informs the range a user is

able to detect a post. Thus, depending of the relative locations of the user and the posts, only some posts may be detected by the user.

Two other attributes, *sync* and *latency*, enable the refining of user behaviour in the environment. These attributes store the synchronization interval to communicate with the server side to send it the user location. The correct knowledge about the Information of user location will determine whether the user may detect a post or not, taking into account the fact that the user is usually moving. The attribute *latency* stores information about how quickly the user will receive the notifications. Notifications about detected posts may be received immediately or at an interval defined by the user. This attribute tries to avoid an overload of notifications at all times.

Other attributes of the *UserSyncPref* class are devoted to storing user preferences on the mobile application behaviour, for instance, how notifications should be shown or grouped, the number of notifications to be shown, notification expiry, etc.

The type of notifications the user may receive are also determined by the user preferences defined in the *UserAdsPref* class. This class stores the set of ad types wished by the user to be received. The types of ads are defined at two levels using the classes *AdsCategory* and *AdsSubcategory* as shown in Fig. 2.

Categories are defined in a set of predefined types as in any other advertising system. This is an open list managed by the system administrator. Each category is refined in one, at least, or many subcategories. This system, close to the well-known web advertising systems, allows easy and full classifying of any type of ads.

The other main objects of the proposal are the classes *Post* and *Ads*. The class *Post* describes virtual or real posts used to stored ads. Each post belongs to only one user, having a unique identification, and a set of information attributes devoted to its description (*namepost*, *imgpost*, between others). Posts have an associated location stored by the attribute *geopost*, and a set of attributes describing date information (creation, update, expiry, etc.) represented in Fig. 2 by the attribute *dateinfo*.

The attributes *typepost*, *techpost*, *grantusers* and *adsinfo* in the class *Post* of Fig. 2, represent a set of attributes aimed to store information about the post behaviour in the environment, and the type of interaction of the user with the post. Hence, the posts can be defined as public or private, allowing other users to be able to store ads in a post belonging to other owners. Moreover, posts can be virtual or real, and, therefore, they can include a tag or not. Furthermore, a post owner also determines if the post admits user interaction by gestures or not. For instance, the owner may wish that the user always touches a real post with a tag in order to read the post information as well as the ads poster, so gesture interaction would not be allowed because gestures could be performed far from the post.

Attributes *grantinfo* and *adsinfo*, shown in Fig. 2, describe a set of attributes also aimed to represent the behaviour of the user with the posts. Thus, a set of attributes are used to store the user activity allowed for the post. For instance, in public posts users could store ads or not, with only a maximum number of ads, ads without images, ads of predefined categories, etc. In addition, the attribute *rangepost* allows the post owner to define the visibility range of the post. Thus, post ads can only be accessed if the user is in the distance range defined by the post, regardless of the value of the attribute *distance* defined in the user preferences.

We can observe that the set of attributes or characteristics defined by the posts allows us to build pervasive and tailored advertising scenarios, by means of the posts characteristics determining their behaviour with the user interaction.

As Fig. 2 shows, the creation of posts and ads is independent. Users can create posts without any ads associated, and ads not necessarily associated to posts. Ads characteristics are described in the class *Ads* of Fig. 2. Each ad belongs to only one user and it is uniquely identified. Usual information on ads is



Fig. 3. Use of TouchMe and PullMe paradigms. a) Touching a posts for read or publish an ads; b) Touching a posts for read or publish an ads; c) Reading ads by means of a shaking air hand gesture.

considered such as: title, image and a text describing the ads in full. Furthermore, the class Ads store information about important date: creation, update, expiry, etc., and the visible information of the ad owner in order to be shown to other users when the ad is notified.

Finally, other attributes store add information about the tag, so ads can store web address, sets of images, videos, etc., the attribute *spaminfo* being used to specify how this information will be transferred and shown to the users.

5. Interaction model: Air hand gesture, location and NFC

The interaction of the user with the posts can be performed using two different interaction paradigms: “TouchMe” and “PullMe” paradigms (see Fig. 3).

The concept of physical browsing was suggested as a natural way to improve the usability of mobile devices and to enable interaction with digital services associated with real world objects in the environment. The concept of physical browsing supports three user interface paradigms: *ScanMe* (the phone device is used to scan an augmented object), *PointMe* (the user points to objects gathering information) and *TouchMe* (the user touches an object) [20,21].

TouchMe paradigm is proposed thanks to the development of the Near Field Communication technology (NFC). Thus, real world objects can be augmented with a RFID tag storing any kind of information or digital services, and users with a supported NFC mobile phone can gather that information and service from the object simply by touching it. The need for proximity between the object and the phone device guarantee the security of the transference process, allowing the application of this paradigm to a wide range of applications including the secure payment [22].

Gesture based interactions in pervasive systems using augmented scenarios has been widely considered lately, due to it being easy-to-use and quite intuitive. The idea is to replace or to improve the menu navigation and manual configuration of services by simple and intuitive hand gestures with the mobile phone. Similar to the *TouchMe*, *PointMe* or *ScanMe* paradigm, Heiske [23] proposes the *PullMe* paradigm.

PullMe paradigm is based on the gathering of information or services from augmented objects disseminated in smart scenarios by simply performing a gesture of movement of the phone away with a

swift pull-away gesture, respectively. The author describes an application of this paradigm to collect information from billboards. Billboards and phones equipped with RFID are enabled to detect each other, so the users move closer his/her phone to the billboard, detects the billboard by means of its RFID identification and he/she makes a “pull me” movement retrieving the information associated to the billboard. The main drawback to this application is the requirement of proximity. The user needs to touch or bring near his/her phone to the augmented object in order to gather the information from the object, so this approximation does not improve the *TouchMe* paradigm at all with standard NFC based applications.

Borrego-Jaraba et al. [24], extend the application of the *PullMe* paradigm in an application called WingTrapper. In this application, the user, performing a “throwing and catching” movement, is able to capture “Trappies” (gifts or rewards located in any place of the surrounding environment). Current user location and stored “Trappies” locations are considered by the system, as well as measurements of the phone sensors, in order to determine the capture or not of a gift by the user. Therefore, this proposal avoids the need for contact proximity between the user and the augmented object.

In the system described in this paper, *TouchMe* and *PullMe* interaction paradigms have been considered. The interaction model will depend on the post type and its characteristics. While real posts are real world objects existing in a real scenario, virtual posts do not exist physically. Real posts supporting RFID tags allows users to close the post and, by touching the tag with a NFC enabled phone interacting with it, write and reads ads. Thus, real posts support the *TouchMe* interaction model (Fig. 3a).

Furthermore, both real and virtual posts are uniquely identified and they have an associated unique location. The knowledge of both post and user location as well as the consideration of the user preferences allows us to apply the *PullMe* paradigm to the interaction of the users with any kind of post. Two types of interaction can be carried out by the user with the posts using the *PullMe* paradigm, as we will describe in depth in the next section: a) publish ads, and b) reads ads. These interactions are carried out by means of user air hand gestures; the user with the phone device in his/her hand makes an intuitive and non-intrusive movement.

Air hand gestures are detected using the movement sensor of the phone, by means of the accelerometer and compass. Figure 3b shows the gesture modelled for the publishing ads interaction and Fig. 3c shows the gesture modelled for the read ads interaction.

Publish ads interaction is performed with a “throwing” movement. The user with the mobile phone in his/her hand makes a pull away movement to publish the select ad in one of the allowed surrounding posts (Fig. 3b). The permitted posts are determined considering the relative user and post location, the post grants (specified in the attribute *grantusers* of the *Post* class) and the user preferences on activity range (attribute *distance* of the *UserSyncPref* class).

Thus, after the “throwing” movement, the allowed posts are shown to the user in the interface of the mobile application, the user choosing the desired post from the list to publish the ads.

The reading of ads, as we describe in the next section, can be carried out by the user in two ways: a) by advising, and b) by searching.

Reading by advising does not require user hand gestures. Considering the user preferences, user location, location of the posts in the environment and the ads published in the posts, the user receives ad notifications in his/her mobile phone, deciding if he/she wishes to read them or not.

Reading by searching can be also performed without hand gesture, simply using the application functionality defined by the “search” process and/or a mandatory synchronization with the server side, and using hand gesture.

The reading by searching process based on gesture is composed of a simple gesture. Figure 3c shows the searching gesture, in which the user makes a “shake” gesture with the phone in his/her hand in order

to detect the ads and their corresponding post satisfying the search criteria or user preferences. This gesture allows recovery of the list of ads and post matching the search, shown to the user in the interface of the mobile application as new notifications.

The “throwing” movement user for publishing ads is a well-known movement previously used by different applications for transferring information between devices or users. That is an accepted movement needing just the measurement of the accelerometer sensor for its recognition.

The shaking gesture proposed by the searching and reading process is also a simple and non-intrusive movement. The user just only has to shake the hand and the process is initiated. The recognition of this gesture only requires the control of accelerometer sensor. Accelerometer is used for the recognition of a continuous hand movement of at least of 2 seconds.

6. Pervasive functional model

The system server component incorporates all the functionality for the management of ads and user posts. Logged users can manage this functionality from the mobile application as well as a laptop. However, a pervasive system should allow the users to interact with the environment in a more natural way.

Thus, the main functionality of our proposal is to publish ads and to receive, with or without user participation, desired ads from the surrounding real world. The user context defined by the user location determines the environment behaviour, consisting of sending notifications to the user about the desired advertising, as well as allowing the user the publication of advertising in a fully smart augmented environment.

Ads are the core of the system. Users can create ads using the mobile application, these ads being stored in the server database. Ads are independent of the posts, so they can be linked to posts or not.

Posts have an associated location, so the management of posts requires assigning to a post its latitude and longitude position. The GPS post position can be given manually by the user or gathered from the mobile device at the location where the user desires create a post.

Users can assign their ads to their posts using the mobile application interface without needing to be located near the post. Thus, the user selects an ad and assigns it to one or more of his/her own posts. However, users can also assign ads to public posts whether supporting or not NFC tags. If the post has a NFC tag associated, the process is quite simple. The user selects the ads from the mobile application and he/she touches the tag associated to the post, after which synchronization is established with the server and, if the process is allowed, the information is stored in the database.

If the post has no associated NFC tag, that is, it is an outdoor public virtual post, the user can use the web or mobile applications by selecting the desired posts to perform the process.

As observed in the sequence diagram of Fig. 4 (top), the first step is the location of the post using the searching process. This process can be performed using shake gesture or by means of the search function, gathering the list of posts located within the range of distance to the user location established by the user in its preferences. Next, the user selects the desired post from the list and, using a throwing gesture or an option of the mobile application, assigns the ads to the post. Once the public post has associated and user ads, the post is stored as user post not requiring the user to be near the post for later assignation of new ads or to delete the ads from the post.

Furthermore, the pervasive advertising system proposed in this paper is based on a tailored ads notification system. Based on user location and user preferences, the system sends notifications to the user corresponding with ads that fulfil user preferences.

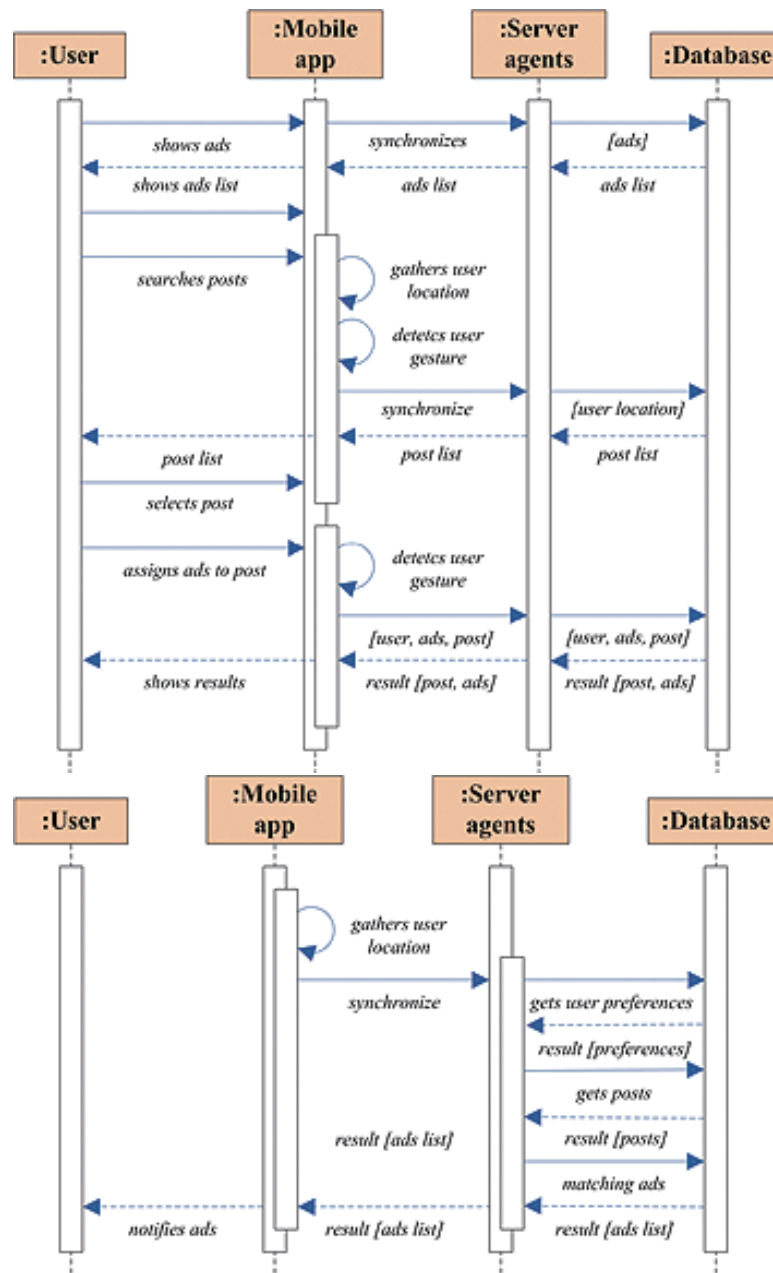


Fig. 4. Sequence diagram of the assignment of ads to posts (top) and ads notification process (bottom).

This process is performed without user participation as is shown in Fig. 4 (bottom). Mobile application gathers user location at time intervals defined in the user preferences, synchronizing with the server. Next, server agents devoted to this process retrieve user preferences and the posts fulfil the distance range defined by the user. Finally, a matching between the ad categories stored in the database for the retrieved posts and user preferences is carried out, obtaining the ads to be notified to the user.

6.1. An efficient implementation of the pervasive notification process

The pervasive information system (PIS) [25] proposed in this paper requires an efficient implementation in order to provide a quick response to continuous user requests. As described in previous sections, the system should notify users about desired ads at the time the user is moving in the real world. Thus, in a real scenario, users go passing from one real or virtual post to another, and the system must gather the current user location and match user preferences about ad categories, with the ad categories associated to these posts closed to the user.

These functional characteristics imply the design of efficient data structures in order to store user preferences, posts and ad information, allowing for fast data retrieving given a high or very high number of requests.

Thus, ads categories and subcategories have been encoded in a fingerprint data representation. Fingerprints are binary array structures used for solve a wide type of problems in computational applications. With this structure an object or entity is represented by a binary array of a predefined size. Each fingerprint element takes values 1 or 0, representing the presence or absence of a property or characteristic in the represented entity.

We have defined a 512 bits fingerprint, large enough to represent the set of all possible ad categories and subcategories. Thus, user preferences about ad notifications are stored in a fingerprint structure, where bits set to 1s represent a user desired category/subcategory. The encode/decode process is carried out by a devoted agent on the server side, updating the user fingerprint structure stored in the server database at the same time the user updates this information using the mobile application.

In addition, a fingerprint structure is also used to represent the category/subcategory associated to ads. Thus, *PostAds* object is defined in the server database to represent the relationship between ads and posts, storing the following information: a) post identification, b) ads fingerprint.

The fingerprint structure is composed of different slices. A first slice is used to represent the different categories. A bit set on in this slice represents the existence of at least one bit set on in the slice corresponding with the subcategory. Therefore, different fingerprint slices are also used to represent all the subcategories corresponding to a category. Bits set on in a subcategory slice represent the presence of a desired subcategory for a user or an ads.

6.2. An efficient implementation of large geolocated database

Information about the posts is another complex problem to solve in an application where the number of posts can be huge, and it is necessary in real time to know the posts in the activity range of many different users. The activity range of the user is defined by the attribute *distance* of the *UserSyncPref* class (see Fig. 2), so ads notifications should be sent to the user for all those posts located within this distance value, taking into account the user location and the post location.

Post location is known and it is stored in the server database using the attributes *latitude* and *longitude* of the *Post* class. User location is gathered by the mobile application at an interval defined by the attribute *latency*, and this information is sent to the server at an interval defined by the attribute *sync*, both of the *UserSyncPref* class (see Fig. 2). Thus, when the synchronization is made, the server needs the calculation between the last user location and posts' locations, selecting all posts within the user activity range.

Then, the process will consist of the calculation between two points (user location and post location) a high number of times (the number of posts can be very large). In order to diminish this calculation it is convenient to diminish the number of comparisons, selecting a subset of appropriate posts. The solution used has been to define points of reference (*PoR*).

A *PoR* is a virtual post system defined for referencing other closest posts by means of a 2D point on the meter scale. In order to have a high performance in the retrieving process we have meshed the locations where posts can be created. Thus, database structures storing information about the countries, cities and towns and for each town a square meshed has been built.

In this mesh creation process, a *PoR* is defined as the centre of each mesh, the number of meshes being determined by the size of the meshed (different for each two) and the distance between the *PoRs*. In most cases, this distance has been taken to 500 meters. In this way, for instance, for the city of Córdoba (Spain), we have initially created a meshed of 625 meshes in an area of 150 square kilometres. Later, this meshed was refined eliminating meshes without any interest, so a final meshed with 322 meshes was considered in the testing of the proposal described in the next section.

PoRs have assigned location information (latitude and longitude) and they are stored in a database structure containing information about their geographical information (country, city, town, meshed and mesh).

Finally, posts' information includes the reference to the closest *PoR*, in order to improve the retrieving performance. Thus, the data structures defined above allow increasing the speed of the matching process. This process is carried out by a devoted agent on the server side as follows:

- The mobile application synchronizes with the server side when users update his/her user preferences. If preferences on desired ad categories/subcategories are modified, the server side modifies the data structure *UserFingerprint*. User fingerprint view stores user identification and the encoded user preferences.
- The mobile application sends the user location at an interval of the *sync* attribute or when the user performs the search option. User latitude and longitude values are matched to the closest *PoR*.
- Depending on the distance value defined for the user to gather information, the closest matched meshes are recovered, considering the size of the meshed.
- Next, the posts belongs to the recovered *PoRs* are selected, and distance value between the posts values and user location is calculated. Next, posts within the user activity range are chosen from the *PostAds* structure.
- Finally, a binary match between the user preference fingerprint and the ads preferences fingerprint corresponding to the chosen posts is performed, and those ads satisfying the user preferences are notified to the user.

6.3. Tailoring non intrusive advertising

Intrusive advertising is an invasive marketing technique devoted to steering the consumer towards a product or service [26–29]. While different studies determine that the consumer is attracted to a product thanks to invasive and spam advertising, nowadays consumers consider it as bad marketing strategy and bad publicity, and developing an antipathy towards a business, meaning the marketing campaign has accomplished the opposite of its aim. However, the feeling of intrusive ads varies among consumers, so it is the preferences of each user that determines how intrusive or not the ads are.

Our system personalizes the ads received by users throwing different attributes defined in the user preferences. Thus, user defines the latency of notifications, determining the time interval the ads can be sent to the user automatically. These ads should match the preferences defined by the user about the type and ads content.

Moreover, other user preferences determine the frequency that same ads can be sent to the user. Thus, the user can choose to block ads in some ways: a) permanently, then the ads would never be sent again

to the user anyway, b) temporally, so the user determines the interval of time an ad may be sent to the user again.

This information is also stored in the server database, so when a notification with news ads is sent to the user, automatically or by user mandatory, selected ads retrieved considering the user request and user location allocated in the matched posts are filtered regarding to the user ads blocked.

This method avoids spam with repeated and undesired ads to the user, at the same time as reducing the number of notifications to the users when a synchronization of search process is performed.

7. Testing the system performance and reliability

We have carried out the feasibility of the proposed data structures and system functionality carrying out many experimental tests. In this section we show the results for two experimental testing the behaviour of the system for the two different scenarios. Processing times have been obtained using a computer with the following characteristic: Intel® Core™ i5 CPU M 480, 2.67GHz, 6 GB de RAM.

7.1. First use case

7.1.1. Goal

The objective of this test is to check the performance and robustness of the developed data structures and algorithm for gathering the appropriate posts existing in the scenario to a distance given by the user preferences. In this test, we are going to consider that all ads allocated in the gathered posts match the user preferences. Hence, we only consider the performance of the process to recover those posts that would allocate ads to be sent to the user, not the ads content allocated in those posts.

7.1.2. Scenario

The simulated scenario is the city of Córdoba. Córdoba is medium size city of Spain with a population of around of three hundred thousand, an overall size of 576 km². The downtown size is around 16 km² and the centre is located at 37.884722°, -4.778889°.

7.1.3. Posts

In this scenario we have studied the performance of the posts recovery for different number of posts. Thus we have considered posts numbers from 10³ to 10⁶ with both random and Gaussian distribution from the centre point of the selected scenario.

Figure 5 shows the different post distributions for 10³, 10⁴, 10⁵ and 10⁶ posts for a circular post location in the scenario using random and Gaussian distribution for post creation. For all the posts, the attribute *rangepost* has been taken to “*undefined*”, in order that all posts can show their information at any distance from the user, the value of the attribute *distance* in the user preferences being what determines the post visibility.

7.1.4. User locations

Five different user locations and four different values of the *distance* attribute have been taken in order to analyze the performance of the recovery process. User locations are distributed in the scenario area, and we have selected distances of 50, 100, 500 and 1.000 meters for the value of the distance attribute defined in the user preferences.

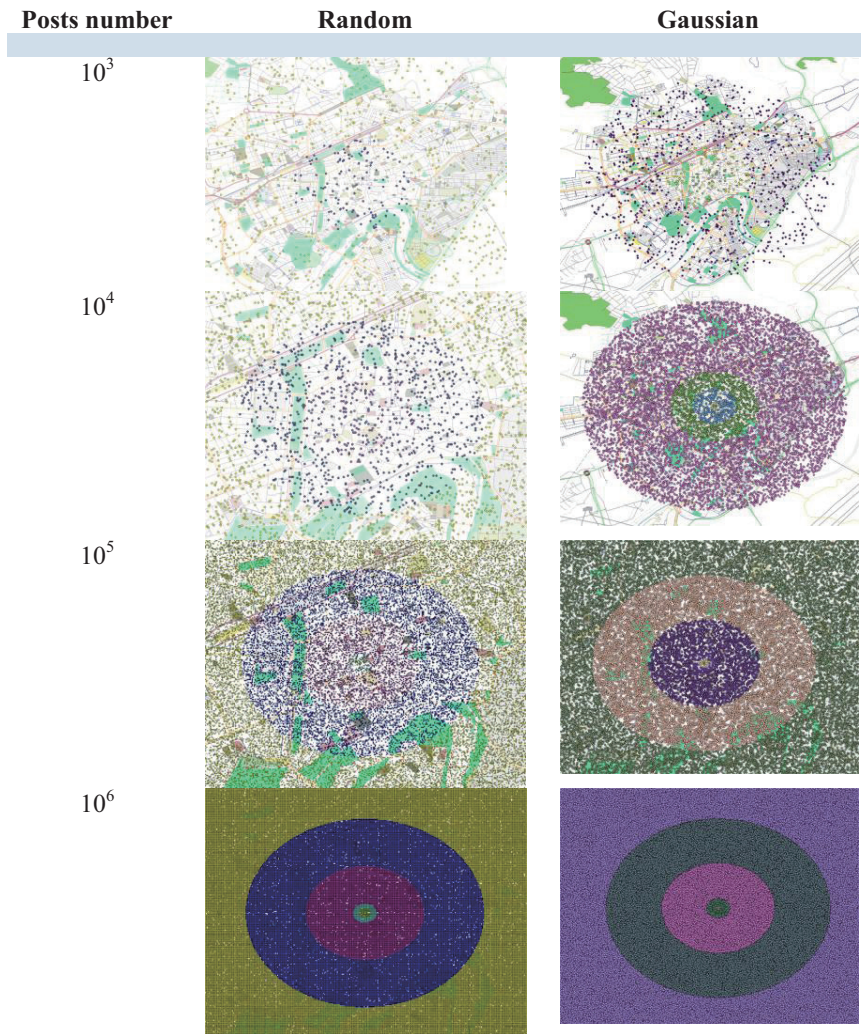


Fig. 5. Scenario for the second test use case.

7.1.5. Results

Table 2 shows the results obtained in the study of the performance in the process of gathering the existing posts in a circular area to a given value of the *distance* attribute defined in the user preferences.

As observed in Table 2, no significant differences exist for random and Gaussian distribution for the same value of the distance and number of posts in the scenario. Although for each distribution the percentage of posts closest to the user location is slightly different, the recovery times for a set of posts are close similar for both distributions for a similar number of posts recovered.

Moreover, the same behaviour is observed for the percentage of error obtained. The percentage of error has been measured as the addition of false positive and false negative; that is, those gathered posts that do not have to be considered and those not gathered posts that should be considered, respectively. We have obtained for all cases a percentage of error less than 0.2% in the worst situations. That is, in the recovering of 10.000 posts, for instance, only 20 posts were badly recovered. The error is always due to those posts located in meshes on the border of the distance value.

Table 2

Study of the recovery performance for random and Gaussian posts distribution. User locations: UL-1: (-4.779536, 37.884466), UL-2: (-4.779407, 37.878868), UL-3: (-4.76448, 37.87217), UL-4: (-4.789524, 37.8885), UL-5: (-4.772224, 37.889088)

	User locations	distance = 50 m		distance = 100 m		distance = 500 m		distance = 1.000 m		
		Posts	Time (ms)	Posts	Time (ms)	Posts	Time (ms)	Posts	Time (ms)	
Scenario with 1.000 posts										
Random distribution	UL-1	1	16	2	16	27	16	101	25	
	UL-2	0	15	0	15	30	16	110	26	
	UL-3	0	15	1	15	31	15	109	26	
	UL-4	0	15	0	15	28	15	116	27	
	UL-5	0	15	2	16	16	15	91	28	
	Scenario with 10.000 posts									
	UL-1	3	22	13	18	290	18	1134	34	
	UL-2	2	21	13	17	282	18	1098	35	
	UL-3	4	22	7	16	270	17	1101	34	
	UL-4	5	22	13	17	284	17	1150	35	
	UL-5	2	22	12	17	261	17	1081	36	
	Scenario with 100.000 posts									
	UL-1	28	40	121	35	2757	40	11034	105	
	UL-2	21	1	105	35	2668	40	11068	112	
	UL-3	32	38	106	33	2667	37	11010	107	
UL-4	30	37	115	31	2672	36	10890	111		
UL-5	26	37	102	32	2824	37	11067	115		
Scenario with 1.000.000 posts										
UL-1	269	360	1086	364	27547	423	111369	1397		
UL-2	273	356	1100	361	27652	417	111250	1480		
UL-3	284	307	1103	312	26820	371	111111	1386		
UL-4	282	306	1130	312	27334	373	110906	1486		
UL-5	257	311	1126	314	27704	374	111248	1526		
Scenario with 1.000 posts										
Gaussian distribution	UL-1	8	17	44	17	673	18	989	30	
	UL-2	1	16	5	16	255	16	829	31	
	UL-3	0	15	0	14	0	14	0	26	
	UL-4	0	16	1	15	45	16	450	29	
	UL-5	0	15	4	15	104	16	620	31	
	Scenario with 10.000 posts									
	UL-1	119	38	441	37	6746	47	9906	82	
	UL-2	17	31	70	31	2425	36	8175	82	
	UL-3	0	21	0	21	0	21	22	52	
	UL-4	1	23	6	23	443	24	4504	73	
	UL-5	7	26	27	26	1092	28	6403	79	
	Scenario with 100.000 posts									
	UL-1	1111	244	4396	247	66796	346	98873	622	
	UL-2	175	182	744	183	24017	223	81910	622	
	UL-3	0	96	0	93	0	92	221	343	
UL-4	21	107	67	109	4405	116	44828	549		
UL-5	53	132	219	132	10578	146	63714	595		
Scenario with 1.000.000 posts										
UL-1	11414	1863	44250	1908	668744	2991	989092	4046		
UL-2	2017	1158	8063	1167	238733	1572	820469	3671		
UL-3	0	18	1	16	5	16	2341	257		
UL-4	152	349	603	349	44630	439	450595	2713		
UL-5	546	569	2261	577	104865	756	635533	3193		

In addition, Table 2 shows the recovery time spent in the process. Obviously, the time increases as the number of posts present in the scenario and the value of the *distance* attribute increases, so also the number of posts to be recovered increases.

However, the time of process is markedly low. For distance value of 100 meters, possibly the most used by the users in proximity advertising search, the time spent by our algorithm is around 1s recovering more than four thousand posts (using a personal computer).

The proposal of the meshed environments is the reason for this low recovery time. If we do not consider meshed scenarios, the time spent on the recovery of a set number of posts increases up to 400% considering the same conditions.

7.2. Second use case

7.2.1. Goal

In this test we check the capacity of the proposal to recover the information allocated to the posts satisfying the user preferences. Posts can be allocated a not predefined number of ads containing information of different categories and, as well, the user has defined preferences on categories of ads he/she wishes to receive.

In this test, we test the performance and correctness of the recovery process for different user locations, values of the distance attribute, number of posts and ads in the scenario.

7.2.2. Scenario

The scenario for the test was the same as that used in the above commented use case.

7.2.3. Posts

Because the behaviour of random and Gaussian post distributions is very similar, in this use case we only show the results for a random distribution of the posts in the scenario. As with the previous use case, we have considered posts number from 10^3 to 10^6 with random distribution from the centre point of the selected scenario.

7.2.4. Ads

We have built 10 ads of different type and content, that is, they have assigned a different value of the attribute *AdsSubcategory*. Only one of these ads matches the user ads preferences, having assigned the same value of the *AdsSubcategory* attribute as that defined in the user preferences. Later, we randomly selected five ads to be allocated in the posts present in the scenario. Therefore, not all posts contain ads matching the user preferences.

Thus, in this use case, the scenario will contain five times more ads than the number of posts, that is from $5 \cdot 10^3$ to $5 \cdot 10^6$ ads, randomly distributed depending on the ad type.

7.2.5. User locations

The same values, as the first use case, of the attribute distance (50, 100, 500 and 1000 meters) defined in the user preferences have been taken.

7.2.6. User preferences

We have considered one value of *AdsSubcategory* attribute (see Fig. 2) in the user preferences, so we will check the type and content of the ad recovery, for the post must match the user defined preferences.

Table 3

Study of the recovery robustness for Gaussian posts distribution and random ads distribution. User locations: UL-1: (-4.779536, 37.884466), UL-2: (-4.779407, 37.878868), UL-3: (-4.76448, 37.87217), UL-4: (-4.789524, 37.8885), UL-5: (-4.772224, 37.889088)

User locations	Distance = 50 m		Distance = 100 m		Distance = 500 m		Distance = 1.000 m	
	Posts	Time (ms)	Posts	Time (ms)	Posts	Time (ms)	Posts	Time (ms)
Scenario with 1.000 posts and 5.000 ads								
UL-1	0	15	1	15	20	15	57	15
UL-2	0	13	0	13	19	15	56	15
UL-3	0	13	0	15	14	15	49	15
UL-4	0	13	0	13	17	15	71	15
UL-5	0	13	1	15	16	15	53	15
Scenario with 10.000 posts and 50.000 ads								
UL-1	1	36	10	36	157	37	656	38
UL-2	0	36	9	36	176	37	636	38
UL-3	2	36	5	36	160	37	630	37
UL-4	7	36	12	36	178	37	651	38
UL-5	1	36	7	36	167	37	630	38
Scenario with 100.000 posts and 500.000 ads								
UL-1	19	251	79	251	1762	259	6814	277
UL-2	11	255	49	251	1689	258	6913	282
UL-3	22	254	58	251	2636	259	6660	284
UL-4	28	253	83	252	2663	260	6784	280
UL-5	15	249	69	249	1693	261	6747	285
Scenario with 1.000.000 posts and 5.000.000 ads								
UL-1	156	2410	611	2416	16837	2486	68049	2671
UL-2	155	2408	686	2426	16926	2498	67767	2694
UL-3	184	2414	703	2441	16276	2499	67382	2681
UL-4	161	2414	679	2427	16763	2489	68077	2687
UL-5	152	2423	678	2427	16816	2498	67667	2686

7.2.7. Results

Table 3 shows the results obtained in this study. Performance of the process is measured by means of the time spent in recovery of the ads satisfying the user preferences and allocated in those posts located in the range of the distance attribute defined by the user (50, 100, 500 and 1.000 meters).

As observed in Table 3, the times spent on the recovery of ad information are very similar to those shown in the first scenario. The data structures used by our proposal allow us to recover the ad information allocated in the posts without a valuable computational cost.

In addition, the process error is very low. The ad error percentage is measured as the addition of false positive and false negative; that is, those gathered ads that do not have to be considered, because they do not match the user preferences, and those not gathered posts that should be considered, respectively.

We have computed that the error is lesser than 0.2% and only error recovery is due to false negative, that is, some posts in the range are not considered due to the fact that these posts are on the border of the distance consideration. As we observe in Table 3 our solution is able to recover in 2 seconds, using a personal laptop (PC Intel i3 with 4 MB Ram) more than five million points and to carry out a proximity matching considering preferences.

8. WingPosts: A publicity ubiquitous system

WingPosts have been developed using different technologies in order to be operable for different mobile operating systems. Technologies such as HTML5 [30] and CSS3 [31] for mobile application



Fig. 6. Some snapshots of the WingPosts mobile application.

development. The PhoneGap framework [32] has been used to access the hardware of mobile devices. For the web server, the framework Ruby on Rails [33] and PostgreSQL databases [34] have been used.

Mobile application can be downloaded from the user web portal and in the installation process the user must register, no anonymous users are allowed by the system, so the user must sign a contract, taking responsibility for the use of the system and its ads content. Because WingPosts use the notifications systems belonging to the operating system of the mobile phone, users should register in the corresponding notifications system. Thus, for instance, for Android mobile phones, the user should have a Gmail [35] mail address.

Figure 6 shows some snapshots of the mobile application. The main screen (Fig. 6a) shows the last ads received by the user in the last synchronization process. Mobile application allows the users to manage their preferences. Users decide what type of categories and subcategories should have the ads allocated

in the posts in order to be received (Fig. 6b). In addition, the user decides the value for other preferences devoted to determining the notification activity. Thus, as shown in Fig. 6c the user assigns values for the attribute distance, latency, number of notifications, style and order the notifications are shown, etc.

Notifications about ads are received by the user automatically by means of notifications at intervals determined by the latency attribute. These notifications correspond to ads allocated in the posts located within the value of the distance attribute defined by user and containing information matching the desired type of ads defined by the user by means of the *AdsSubcategory* attribute (Fig. 6c). However, the mobile application allows the user to change these preferences using a mandatory search. Figure 6d shows a screen of this functionality. Users select the desired ads content writing a text and the desired ads category and a personalized search is processed. The result of the search is a list of ads fulfilling the user preferences, as shown in Fig. 6e. Then the user can choose one ad and visualize the detailed ad information (Fig. 6f).

As shown in Fig. 6f, detailed ad information includes information about the advertiser, so the user can make a phone call, write an email, connecting with the advertiser website or social network, etc., depending on the information included in the ads.

In addition, as shown in Fig. 6g, information about the post where the ad is allocated can be visualized by the user. Touching the ad image or displacing the screen to the right, a map with post location and post information is shown on the screen. Thus, the user can get information about the post creator and even access to its website.

Furthermore, in order to simplify the user interaction, computational gesture is considered for the search process. Thus, by simply shaking the mobile phone, the search is initiated using the defect or defined preferences. This gesture permits expedite and simplifies the process allowing the user to carry out a search quickly without having to wait for the next synchronization time.

Mobile application also integrates all the necessary functionality for users to manage their own ads and posts. Users can visualize their created ads (Fig. 6h) and edit them using a simplified form (Fig. 6i). Posts also can be managed by the user (Fig. 6j), and users can visualize on a map their own posts as well as other closer public posts where the user can allocate one or several ads (Fig. 6k).

Moreover, users can assign ads to posts using a simple screen (Fig. 6l) and also using a hand gesture. Thus, users can select an ad and, simply using a throw away gesture with the mobile phone, the selected ad is allocated to the nearby private posts and nearby public posts. This natural movement is simple and speeds up the quick ads allocation in posts when the user desires public ads in closer locations to his/her actual position in scenario.

Furthermore, WingPosts website (Fig. 7a) includes the full user functionality for users to manage their advertising information. Registered user can create ads (Fig. 7b) including text, images, videos, links, etc. These ads contain short and detailed descriptions (Fig. 7c), so the users can test how this information will be shown to other users in the mobile application.

As well as users managing ads, they can manage posts. Users create posts using an easy and graphic interface based on maps. Users create posts allocating their position in a map (Fig. 7d), and users can delete the posts or to change their position at anytime. The post lists created by the users are shown in the interface and on the map the user can visualize their exact position.

Once the user has defined posts and ads, he/she can allocate these ads in their own posts as well as the public posts (Fig. 7e). Using a form including a map, the user can assign and visualize which of their ads are assigned to one or several of the allowed posts. Moreover, the full managing of ads and post can be performed using forms (Fig. 7f). This form allows the user to create, delete and change information on ads, posts and their relationships.

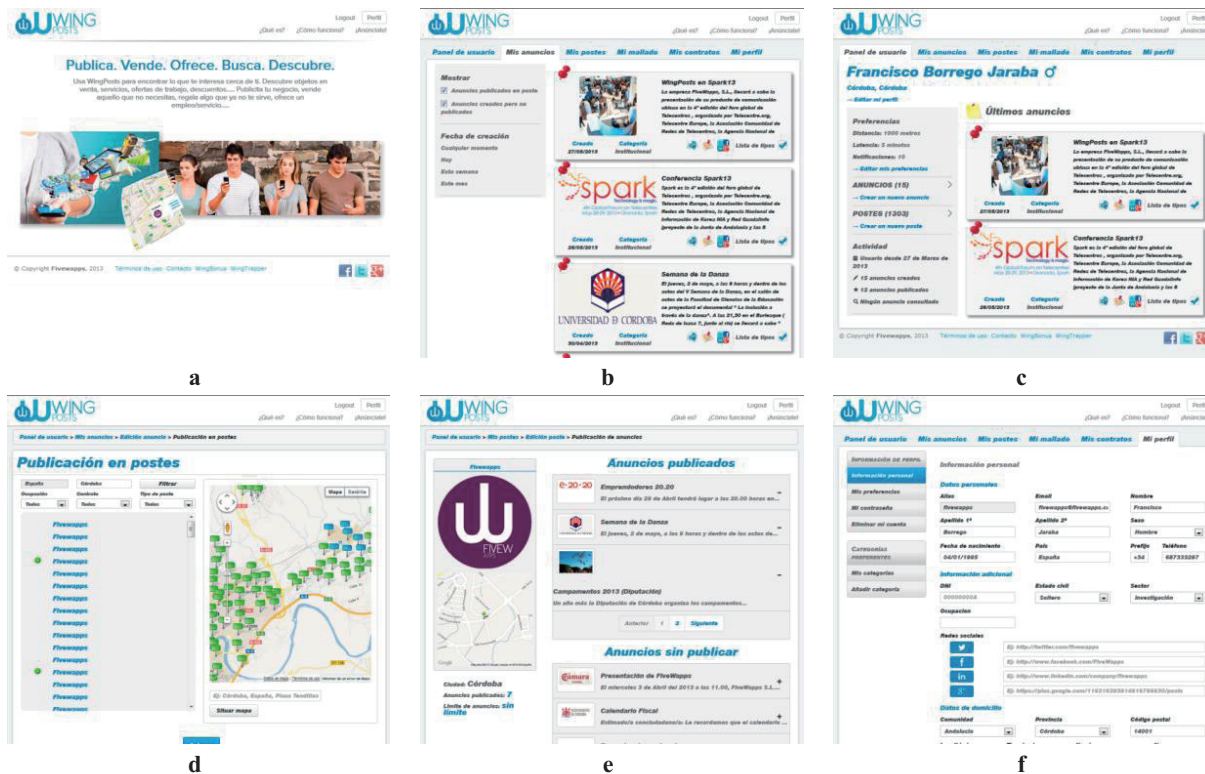


Fig. 7. Some snapshots of the WingPosts website.

The interface of WingPosts website is conceived to be clear and easy to use, taking into account that users can create any number of ads and posts, posts can be assigned any number of ads, users can also assign ads to public posts created by other users and their own WingPosts, etc. Thus, the number of elements that a user can manage can be large and very changeable in the time, therefore a simple interface and map-based is highly recommendable.

9. Discussion

In this paper we have described a very efficient solution for ubiquitous publicity using different mobile technologies.

Our proposal allows users to create cities of virtual totems or posts and to attach to these totems any contents or ads. Posts can be assigned to real posts or not, and ads can include any kind of content, texts, social networks info, images, and so on. In addition, the posts can include multimedia information about the post owner.

Both, Web and mobile application allow users, the full management of posts and ads. Thus, posts can be created, updated and even moved from one side to another of the “world” with just a simple user click. Moreover, users can allocate or deallocate ads to posts by means of a simple click, thanks to a useful and easy mobile and Web interface.

In order to give answers to the problems of the management of a huge number of posts and ads we have designed and built an efficient database. Environments and cities have been meshed. So, posts are

referenced to their corresponding closest points in the nearest mesh. We have tested our database in experimental and real tests and we have demonstrated that we are able to retrieve the information with a depreciable error in less than a second for more than a million points, using a personal computer.

The system is a real and operative system operating in some Spanish cities. Currently, we are working to extend the system to main Spanish and European cities.

Acknowledgment

This work was supported by the Ministry of Science and Innovation of Spain (MICINN): Project: TIN2011-24312.

References

- [1] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson and A. Oliveira, Smart Cities and the future internet: towards cooperation frameworks for open innovation, *The Future Internet* (2011), 431–446.
- [2] N.A. Streitz, Smart Cities, Ambient Intelligence and Universal Access, 6th International Conference on Universal Access in Human-Computer Interaction Context Diversity, *Lecture Notes in Computer Science* **6767**, Springer-Berlin Heidelberg, 2011, pp. 425–432.
- [3] N.K. Ché, N. Pardons, Y. Vanrompay, D. Preuveneers, and Y. Berbers, An intelligent domotics system to automate user actions, *Adv Intell Soft Comput* **72** (2010), 201–204.
- [4] A. Olaru and A. Florea, Context-aware agents for developing AmI applications, *Journal of Control Engineering and Applied Informatics* **13** (2011), 42–50.
- [5] M. Weiser, Ubiquitous Computing, *IEEE Computer* **26** (1993), 71–72.
- [6] C. Ramos, J.C. Augusto and D. Shapiro, Ambient Intelligence: The next step for artificial intelligence, *IEEE Intelligent Systems* **23** (2008), 15–18.
- [7] ECMA. Near Field Communication white paper. ECMA/TC32-TG16/2004/1, Retrieved: May, 2013.
- [8] H.E. Byun and K. Cheverst, Utilizing context history to provide dynamic adaptations, *Applied Artificial Intelligence* **18** (2004), 533–548.
- [9] M. Duckham and L. Kulik, Location privacy and location-aware computing, in: *Dynmaic and Mobile GIS: Investigating Change in Space and Time*, J. Drummond, ed., Boca Raton CRC Press, 34–51.
- [10] WingPosts by FiveWapps, S.L.TM. Available online: <http://www.wingposts.com> (accessed on July 2013).
- [11] P. Holleis, G. Broll and S. Böhm, Advertising with NFC, *The Eighth International Conference on Pervasive Computing* (2010), 1–10.
- [12] U. Varshney, An Approach for smart artifacts for mobile advertising, in: DESRIST'12 Proceedings of the 7th International Conference on Design Science Research in Information Systems: advances in theory and practice (2012), 147–151.
- [13] S. Oumtrakul, N. Chanuntawaree and J. Gao, AdTouch: A 2D-Barcode Mobile Advertising Service System, *Security-Enriched Urban Computing and Smart Grid* (2011), 188–202.
- [14] L. Aalto, N. Göthlin, J. Korhonen and T. Ojala, Bluetooth and WAP push based location-aware mobile advertising system, in: *MobiSys'04 Proceedings of the 2nd International conference on Mobile Systems, Applications and Services* (2004), 49–59.
- [15] H. Haddadi, P. Hui and I. Brown, MobileAd: private and scalable mobile advertising, in: *MobiArch'10 Proceedings of the Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture* (2010), pp. 33–38.
- [16] A. Dey, G. Abowd, P. Brown, N. Davies, M. Smith and P. Steggles, Towards a better understanding of context and context-awareness, in: *HUC'99 Proceedings of the 1st International Symposium on Handheld and Ubiquitous, Computing* (1999), pp. 304–307.
- [17] H. Martins and N. Silva, Characterization, Comparison and Systematization of Context Ontologies, in: *CISIS'12 Proceedings of the sixth International conference on Complex, Intelligent and Software Intensive Systems* (2012), pp. 983–988.
- [18] P. Benou and C. Vassilakis, The conceptual model of context for mobile commerce applications, *Journal Electronic Commerce Research* **10** (2010), 139–165.
- [19] JCDecaux Group. Available online: <http://www.jcdecaux.com> (accessed on May 2013).
- [20] H. Ailisto, L. Pohjanheimo, P. Välikynen, E. Strömmer, T. Tuomisto and I. Korhonen, Bridging the physical and virtual worlds by local connectivity-based physical selection, *Personal and Ubiquitous Computing* **10** (2006), 333–344.

- [21] E. Rukzio, K. Leichtenstern, V. Callaghan, A. Schmidt, P. Holleis and J. Chin, An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning, *Lecture Notes in Computer Science* **4206** (2006), Springer-Berlin Heidelberg, 87–104.
- [22] G. Madlmayr, J. Langer, C. Kantner and J. Scharinger, NFC Devices: Security and Privacy, in: *ARES'08 Proceedings of the Third International Conference on Availability, Reliability and Security* (2008), pp. 642–647.
- [23] S. Hinske, Pulling Digital Data from a Smart Object: Implementing the PullMe-Paradigm with a Mobile Phone, *Lecture Notes in Computer Science* **4551** (2007), 306–310.
- [24] Francisco Manuel Borrego-Jaraba, Francisco José Bellido Outeriño, Irene Luque Ruiz and Miguel Ángel Gómez Nieto, Mobile Solution Using NFC and In-air Hand Gestures for Advertising Applications, *International Conference on Consumer Electronics* (submitted).
- [25] S. Najjar, M.K. Pinheiro, C. Souveyet and L.A. Steffene, Service Discovery Mechanism for an Intentional Pervasive Information System, in: *IEEE 19th International Conference on Web Services* (2012), pp. 520–527.
- [26] Y. Truong and G. Simmons, Perceived intrusiveness in digital advertising: strategic marketing implications, *Journal of Strategic Marketing* **18** (2010), 239–256.
- [27] K. Wehmeyers, Mobile Ad Intrusiveness: The effects of message type and situation, in: *20th Bled eConference eMerge: Merging and Emerging Technologies, Processes and Institutions* (2007), pp. 758–775.
- [28] Y. Truong, R. McColl and P. Kitchen, Practitioners perceptions of advertising strategies for digital media, *International Journal of Advertising* **29** (2010), 709–725.
- [29] R. Wei, H. Xiaoming and J. Pan, Examining user behavioural response to SMS ads: Implications for the evolution of the mobile phone as a bona-fide medium, *Telematics and Informatics* **27** (2010), 32–41.
- [30] HTML5. <http://www.w3.org/TR/html5/> Access date: April 2013.
- [31] CSS3. <http://www.w3.org/TR/CSS/> Access date: April 2013.
- [32] PhoneGap Framework. <http://phonegap.com> Access date: April 2013.
- [33] Ruby on Rails Framework. <http://rubyonrails.org> Access date: April 2013.
- [34] Object-Relational Database Management System (ORDBMS) PostgreSQL. <http://www.postgresql.org> Access date: April 2013.
- [35] Google Mail (Gmail). <https://mail.google.com> Access date: April 2013.

Francisco Manuel Borrego Jaraba, is member of the Department of Computing and Numerical Analysis at the University of Córdoba in the area of Computer Science and Artificial Intelligence. Systems Technical Engineer in 2006 and Computer Engineer in 2009, is currently a PhD student. His research focuses on RFID and NFC technologies, smart environments, pervasive computing, context-aware computing, ubiquitous computing and solution development in teaching innovation.

Irene Luque Ruiz, is full professor of Computer Sciences and Artificial Intelligent and she is member of the Department of Computing and Numeric Analysis at the University of Cordoba. Bachelor in Chemical Sciences (1981) and Doctor Lecturer in the Computer Sciences (1995), she is manager in the Software and Knowledge Engineering and Databases research group. Her teaching responsibilities are software engineering, information systems and computational systems analysis and design. His research activities are oriented to smart environments, pervasive computing, context-aware computing, teaching innovation and new computational solutions in Chemistry, between others.

Miguel Ángel Gómez-Nieto, is full professor of Computer Sciences and Artificial Intelligent and member of the Department of Computing and Numeric Analysis at the University of Cordoba. Bachelor in Sciences (1980) and Doctor Lecturer in the Chemical Sciences (1982), since 1990 he belongs to the Software and Knowledge Engineering and Databases research group. His teaching responsibilities are databases, software engineering and computational systems analysis and design. His research activities are oriented to smart environments, context-aware computing, ubiquitous computing, knowledge representation, clustering, graphs, between others.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

