Users as Smart Sensors: A Mobile Platform for Sensing Public Transport Incidents*

Cristian Tanas and Jordi Herrera-Joancomartí

Universitat Autònoma de Barcelona
ctanas@deic.uab.cat, jordi.herrera@uab.cat

Abstract. Sensor networks may become a key element in a smart city in order to collect and provide information to its citizens. In this paper, we propose a new mobile phone sensing application, *Incidències 2.0*, that helps users notify and stay informed about the incidents of the public rail network in the Barcelona metropolitan area. The application takes advantage of the widespread use of smartphones combined with their sensing capabilities to gather sensory data from the environment and then send the sensed information back to a central data collection facility using cellular network technology. Data retrieved from the application provided by real users allows us to make a first analysis on the potentials of this new sensor network paradigm.

1 Introduction

Smart cities use technology and network infrastructure to improve economic and political efficiency and enable social, cultural and urban development. Though there are many factors involved within a smart city, their citizen engagement and the necessity of sensors to monitor the city's activities are becoming key concepts towards a successful deployment.

There is an undeniable need for sensor networks in a smart city to collect and provide information to its citizens. Sensor networks have become one of the most active areas in networking research over the last decade, providing overwhelming potential for information collection and processing in a wide range of environments. The state of the art approaches in sensor networking include a limited number of static devices, usually wirelessly connected, spanned over a pre-determined geographical area gathering evanescent information of the environment surrounding them.

Nevertheless, we can not overlook the increasing popularity and huge potential of smartphones to build a new generation of sensor networks, targeting daily life activities of individuals and the environment surrounding them. Indeed, modern smartphones besides being sophisticated computing platforms, include a wide range of capabilities, like computing (CPU, data storage,...), communication (UMTS, WiFi, Bluetooth) and sensing (positioning -GPS-, motion -accelerometer-, image -camera- , audio -microphone-). In addition, smartphones development is exploding, and competition between Apple and Google expands over 74% of the market share with approximately 149 millions of devices sold during the 4th quarter of 2011 according to Gartner, Inc. [1].

* This work has been partially supported by the Spanish Government through project TIN2010-15764 N-KHRONOUS and the UAB grant PIF 472-01-1/E2010.

J. Nin and D. Villatoro (Eds.): CitiSens 2012, LNAI 7685, pp. 81–93, 2013.

Therefore, we can take advantage of the widespread use of smartphones combined with their sensing capabilities to gather sensory data from the environment and then send the sensed data back to data collection facilities using cellular network technology. Furthermore, it might be useful to have individuals participating in the sensing tasks. Surrounding environment detection, information processing, or great communication skills are just some of the qualities that individuals posses. Therefore, we can take advantage of both available sensors in a smartphone and the smartphone's owner intelligence to acquire better knowledge on long-lasting features of the landscape. Users can provide additional information to sensor readings, such as natural language description of the environment or location-tagged images, thereby provisioning researchers with a substantial wealth of data. When relying on users to act as sensors we could refer to them as *smart sensors*, and we will refer to this type of sensor networks as *smart sensor networks* (SSN).

SSN can help overcome many of the limitations of existing proposals in wireless sensor networks, which require physical deployment and customized node management, in addition to complex communication protocols. However, the new opportunities and benefits offered by modern smartphones as sensing devices come at a price. Bringing together geographically and sociologically unrelated individuals to create a community that performs tasks for a greater good brings up front new challenges and security issues that might have a strong impact on the overall performance of the network. Sensor network managers, now have to deal with potential sabotage (intentional or unintentional) from the smartphones users. How to derive trust in the sensor readings provided by a crowd of volunteer individuals becomes an important research question in these environments. Moreover, to engage as many users to participate in the sensor network's sensing tasks is a major challenge since usually, device owners are reluctant to share their valued resources if no direct benefits are perceived.

In this paper, we present *Incidències 2.0*, a SSN application[1] that allows users to notify and keep abreast of any incident that affects the rail public transport network nearby Barcelona. Although the application is built on top of a general framework that may allow more general sensing tasks, we would like to evaluate the correctness of our framework by developing a real, specific and useful application for that framework that will provide us with real user data in order to analyse the possibilities of a real deployed SSN.

The paper is organized as follows. In Sect. 2, we review the existing proposals in which users take part as sensors entities. Section 3 introduces a new sensing application, identifying its main functionalities. In Sect. 4 we present the architecture and modular design of the proposed framework. The data obtained from the proposed platform is analysed in Sect. 5. Finally, Sect. 6 concludes the paper.

2 State of the Art

Smart sensor networks have a large number of potential applications. However there are just a few proposals that leverage the idea of having sensing tasks relayed to consumer-owned smartphones, and the majority were developed for experimental purposes.

The SSN application spectrum ranges from CO_2 emission monitoring [7] to patient health monitoring systems where smartphones are used in combination with wireless

[1] `http://www.incidencies.org`

(bio) sensors to monitor a patient's vital signs [10], passing through a longer list of location-based services, such as traffic accidents detection and situational awareness provisioning to first responders [15], traffic conditions monitoring [13], or real-time trail network update for hikers and mountaineers [14]. In addition, a built-in GPS receiver and an accelerometer can be used to identify the transportation mode of an individual (i.e. walking, running, biking, or in motorized transport), as described by Reddy et al. [12].

Furthermore, smart sensor networks can provide support in emergency scenarios or environmental disasters as A. Gahran explained in an article on how citizens living in the Gulf Coast region could use their smartphones sensors, such as GPS and cameras, to enter data on the ecological impact of the Gulf oil spill, providing specialists with first hand information of this disaster [8]. This information was latter used to generate impact analysis and provide recommendations.

As for general purpose urban sensing network architecture, the MetroSense project [3] is worth mentioning. MetroSense offers a network architecture for urban-scale people-centric sensing, leveraging existing urban infrastructure and human mobility to opportunistically sense and collect data "about people and for people". Some applications include BikeNet [5] and SkiScape [4], developed as sample studies to demonstrate the usefulness of the platform.

Practically all proposals in people-centric sensing applications face the problem of data reliability. Although some approaches have been studied, such as game theory-based mechanisms, where data pollution detection is combined with punishment strategies [2,11], or entropy dynamics measuring in a descriptive distribution over the course of a game [6], they all assume that one can infer a relationship graph among the members of the sensor network, or there exist a tamper proof hardware providing a "ground truth" for data validation. Alternatively, reputation-based strategies seem to provide a promising solution. Nevertheless, all the sensing applications studies fail to provide a robust data validation mechanism.

3 Application Overview

All existing proposals in smart sensor networks and related areas, such as ad hoc networks, lack a model of users behaviour. Instead, they all assume that users will behave in a pre-determined manner, and they measure the network's performance, or make hypothesis based on this assumption. However, individuals are normally passionate and many times act in an unpredictable way. In order to break with this tradition, we developed a sensing application and a framework to facilitate the implementation of a real-time incident reporting system focused on the rail network services of the metropolitan area of Barcelona, where users in possession of a smartphone running Google's Android or Apple's iOS will act as smart sensors and information providers. Although the application is focused on the rail network services, it is developed on top of a framework that is highly scalable, allowing the implementation of an incident reporting system in other environments, such as traffic or urban furnishing damages.

Fig. 1. Incidències 2.0 common features

The application follows a client/server paradigm in which end-users will be provided with a smartphone sensing application that will provision them with a tool to easily notify the occurrence of a new event, such as a delayed train. Once the user enters all the information, it will be transmitted using the smartphone's network connection to a central data collection facility. Then, the information passes through a validation process, it is stored into the database and is made publicly available to all the devices having the sensing application installed.

3.1 Incident Notification

Incidències 2.0 allows users to notify new incidents and stay up to date of the incidents that are currently going on, through the client application installed on their smartphones.

From the main menu of the client application, users are allowed to report a new incident as shown in the left-hand side of Fig. 1. If a user chooses to report a new rail network incident, he or she must provide information about the rail network service and station where the incident is taking place, the event that caused it, an evaluation of the incident's severity. Optionally, a textual description can also be added. On the other hand, if the user chooses to report an incident of any other type, he or she will be asked introduce a description, which is now mandatory, and make an evaluation of the event's severity. Once the user introduced all the information, it is sent jointly with the current date and time and the user's GPS position, if available, to the Incident Management Center (IMC).

In addition to report a new incident, users can confirm incidents already reported by other participants in the sensor network. In the confirmation process, users are allowed to specify an optional comment or description, which is included in the incident information and can be later viewed in the detailed description of the incident.

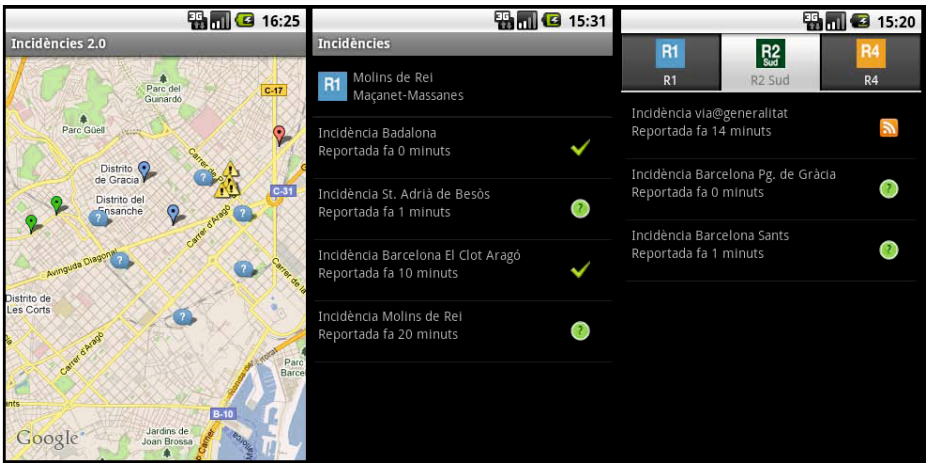

Fig. 2. Incidències 2.0 incidents visualisation options

3.2 Incident Visualisation

Incidències 2.0 provides to all application users the information of all the incidents that are taking place in a precise moment of time, offering various visualisation modes through the client application that the users have installed on their smartphones (Fig. 2).

Incidents can be seen as markers on a map where different colors and different icons are used to distinguish between different types of incidents. Users can select a map marker by tapping on that marker, which will open an alternate menu that offers the possibility to access more information about that incident, or to upload new information regarding it. This feature provides users with a quick and effective way of consulting current incidents in a given region. However, users that usually travel using the rail network system, tend to take just one railway line or a combination of a few of them. Thereby, they should be able to filter the existing incidents according to their preferences. To satisfy this requirement, *Incidències 2.0* allows users to select a specific railway line and see only the incidents affecting that line as a list (central part of Fig. 2). Tapping on one incident in the list will bring to front a detailed view about that incident, from which the user can upload new information or consult the incident's location on the map. If the user combines different railway services or lines, the application gives the possibility to define up to three favourite lines to follow (right-hand side of Fig. 2), so that the user can have a quick access only to the information that might have an impact on his or her quotidian travels.

4 Application Architecture

As we have already mention, *Incidències 2.0* has been developed through a more general framework that follows a client/server architecture with a smartphone sensing application on the client side and the Incident Management Center as the server application.

4.1 The Smartphone Sensing Application

We have developed a sensing application that runs on both Android and iOS-based devices. It implements a data gathering module, which relies on the smartphone's owner as a sensor using its GPS receiver, the current time and date, and a data visualisation module, which connects with the IMC to retrieve all existing incidents.

The Android version of the application was developed using the Android SDK and the Java programming language, while the iPhone version was developed as a native iOS application, based on the Cocoa Touch framework and using the Objective-C programming language. Both versions interact with the device's localisation services to retrieve the smartphone's current GPS position, and with the Google Maps API to offer a map-based visualisation service. Furthermore, the application takes advantage of the device's Internet connection to send the collected data back to a central server.

4.2 The Incident Management Center (IMC)

The IMC is the central part of the framework and it is responsible of processing, validating, and storing the incident notifications provided by end-users. It follows a modular design and it is composed of the following five modules (interrelated as shown in Fig. 3):

1. *Incident Definition Module (IDM)*
2. *Incident Reception and Triage Module (IRTM)*
3. *Data Validation Module (DVM)*
4. *Public Relationship Module (PRM)*
5. *Data Storage Module (DSM)*

Incident Definition Module (IDM). A type of incident is described by an XML codification schema and a set of attributes (i.e. available information regarding the incident) and actions associated with the incident. Although at present time, the application framework focuses on incidents in the rail network services, it is able to deal with new types of incidents through a request sent to the Incident Definition Module specifying the XML codification schema of the new type of incident, along with the attributes and actions associated with it.

Incident Reception and Triage Module (IRTM). The IRTM is the front-end interface of the IMC, allowing end-users to communicate with the central servers to report a new incident or retrieve the existing ones. It also performs a triage phase to check if the type of incident reported or requested is supported by the platform, that is, if it has been previously registered through the Incident Definition Module. In addition, this module is responsible for delivering information about on going incidents, as requested by end-users, and for managing confirmations.

After the received notifications passed the triage stage, the information is forwarded to the Data Validation Module, where it runs through a data validation process.

Data Validation Module (DVM). The DVM provides a validation scheme for the incoming notifications based on the users' reputation and collective knowledge. However, how to derive trust from the information collected by a crowd of volunteer individuals is a

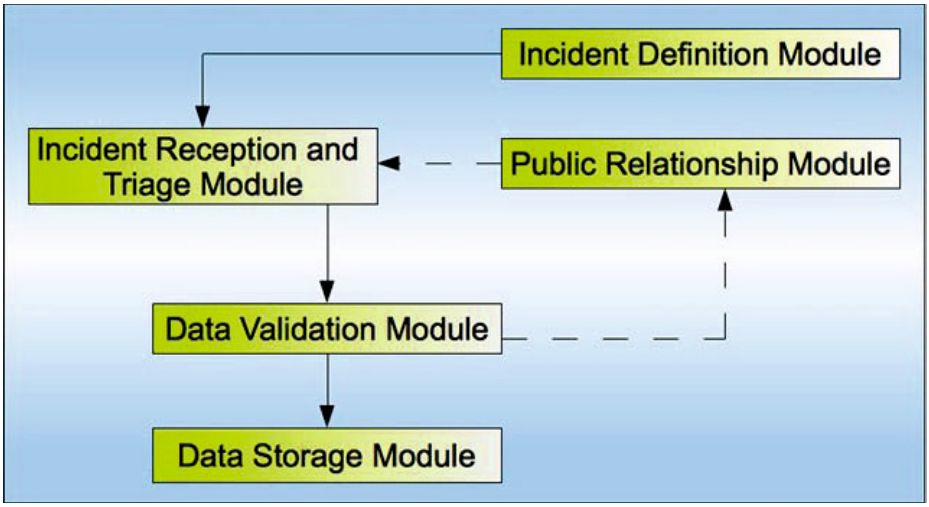

Fig. 3. Modular design of the Incident Management Center

major challenge in people-centric sensing applications. It is not straightforward to assess if a notification received from a user is valid and corresponds to a real event happening at that precise moment, or on the contrary, it provides dishonest or counterfeit information. In Subsect. 4.3 we provide some ideas on how to achieve such data validation.

Public Relationship Module (PRM). *Incidències 2.0* relies on end-users to provide incident notifications and this information is made publicly available for all the other users having the application installed. However, by nature, individuals are selfish and they are not inherently motivated to collaborate in the sensing tasks unless direct benefits are perceived from their participation. Then, selfish users will only consult currently active incidents, but will refuse to provide incident notifications. Therefore, the PRM must provide and manage the incentives that should be provided to end-users to stimulate their cooperation. The PRM could certainly exploit data from the IRTM or the DVM, such as reputation or user behaviour and query patterns to design cooperation protocols that better adapt to the user's needs.

Data Storage Module (DSM). After the incident notifications are processed and validated, the information has to be stored in a database, so it can be retrieved and used in the future by the other modules, or statistically analysed. The database scheme was implemented following a relational database model and using the MySQL database manager system. The design respects a modular scheme, so that new types of incidents can be easily incorporated and stored into the database.

The information stored into the database includes users' credentials and reputation value, the different incident notifications and confirmations, and also the queries performed by the users. In addition, the database must reflect the results of the validation process, results that are stored jointly with the incident notification scheme.

4.3 Reliable Data Readings

Data validation can be performed through a reputation system and collective knowledge to ensure the reliability of the incident notifications sent by the users. The user's reputation provides a measure of his or her credibility within the system. Thereby, the incident notifications received from users with a high reputation value, exceeding a given threshold, will be considered as valid. On the other hand, if the user does not benefit from high credibility within the community, different observations from different users of the same incident will help establish the validity of the incident if a pre-defined number of observations is accounted. Collective knowledge is handled by the application framework through the client application, offering the ability to confirm incidents previously notified by other users. Therefore, if a pre-define threshold of confirmations is reached, then the incident will be considered as valid, and at the same time the reputation of the users that observed that incident will be increased.

At the beginning, each user starts with a neutral reputation score. They can increase their reputation by participating in the validation process of an incident, either notifying the incident or confirming it.

Computing User's Reputation Scores. We need to quantify the reputation of a user and attune the threshold from which a user is considered to have a sufficiently high reputation value to be granted with total credibility. The past interactions with a given user determine up to a certain degree the future behaviour of the user. For example, if a user always reports valid incidents, it is most likely that the next time it reports an incident it would be a valid one.

We take a Bayesian system approach as described in [9]. The reputation score can be computed based upon the beta probability density function parameter tuple (α, β), where α and β represent the valid incident notifications sent by an user and the unconfirmed ones respectively. The beta PDF $f(p|\alpha, \beta)$ can be expressed using the gamma function Γ as:

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\alpha)} p^{\alpha-1}(1-p)^{\beta-1} \quad , \tag{1}$$

where $0 \leq p \leq 1,\ \alpha > 0,\ \beta > 0$.

The probability expectation value of the beta distribution is given by:

$$E(p) = \frac{\alpha}{\alpha + \beta} \quad . \tag{2}$$

When there is no information about the past action of a certain user, the a priori distribution is the uniform beta PDF with $\alpha = 1$ and $\beta = 1$. Then, if r valid incidents and s unconfirmed incidents are observed, the a posteriori distribution is the beta PDF with $\alpha = r + 1$ and $\beta = s + 1$. The modeled PDF expresses the uncertain probability that in the future the user will send valid incident notifications. For example, if a user sends 7 valid incident notifications and 1 that can not be verified, the probability expectation value according to (2) is $E(p) = 0.8$. This can be interpreted by saying that the relative frequency of reporting a valid incident notification in the future is somewhat uncertain, and that the most likely value is 0.8.

However, it is effortful to compute the threshold value for a high reputation score for one user since it strongly depends on context-dependant factors, such as the number of

previous incident notification received from the users or the total number of incident notification in the system. One possible solution would be to publish the incident together with a reputation index associated with the user who notified it. Nevertheless, this approach requires a further explanation for users about the meaning and usage of the reputation score.

Collective Knowledge Management. The validation process of an incident notification received from a user with a lower reputation value than the defined threshold, must combine confirmations from other users, and also notifications that refer to the same incident. Counting confirmations is straightforward since these refer to an existing incident in the database. However, in order to verify that different notifications correspond to the same incident we must verify that those notifications make reference to the same public transport service, the same railway or subway line, the same station, and they were caused by the same event. The necessary information to determine if two notifications correspond to the same incident is completely dependant on the type of incident, and the characteristics that make two notifications refer to the same incident object must be passed along with the incident type definition.

When a new incident notification is received, the first step is to check if it makes reference to an existing incident in the database. If the result from the previous query is positive, then we must verify if the notification corresponds to an unconfirmed incident or to a valid one. In the former case, we have to add the notification to the number of confirmations of this incident and validate it if the number exceed the pre-define threshold. In this case, we must also update the user's reputation accordingly. In the later case, we will increase the number of confirmations by 1 and update the user's reputation, but no change will take place in the incident's state. In addition, the validation scheme must take into account that all incidents have a limited lifetime or time-to-live (TTL). So, only those incidents that have a positive remaining TTL must be considered to review if an incoming notification corresponds to an existing incident in the database.

As in the reputation case, it is hard (if even possible) to determine a threshold value for the number of confirmations required to validate an incident since it strongly depends on context-dependant parameters, such as the total number of smart sensors or the density of users around the incident's area. Once again, a possible solution would be to publish the number of confirmations associated with an incident and let the user decide the validity of the information. As in the reputation score case, further explanation for users is required regarding the meaning and usage of the added information.

5 Application Data Analysis

Incidències 2.0 is currently in use in the Barcelona metropolitan area and the obtained data allows us to draw some analysis regarding the usage of such platform.

5.1 General Application Usage

At present time, the application has more than 3400 users registered in its database and they have performed more than 25000 queries since February the 1st, 2012. Although the application is freely available both in the Android market and in the Apple Store,

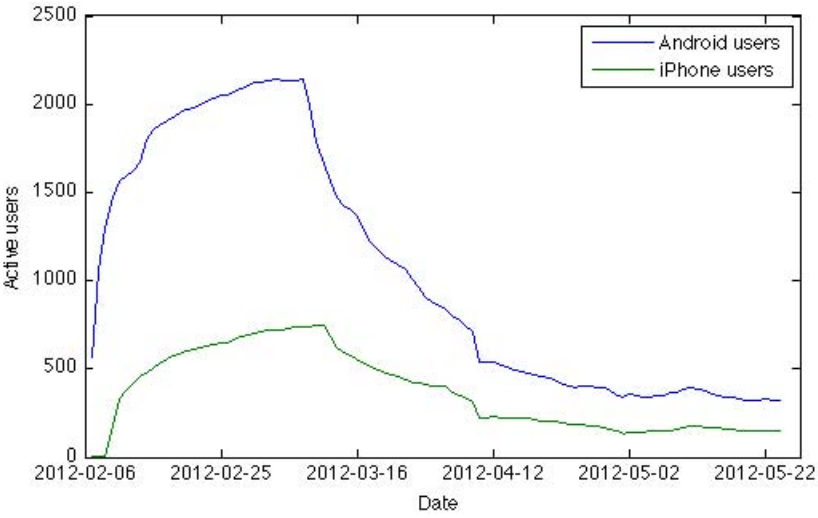

Fig. 4. Number of active users by operating system

the user distribution between platforms is not uniform, having a $74,7\%$ of users using an Android device and $25,3\%$ an iPhone.

In order to properly analyse the time evolution of the application usage, we define an *active user* at a particular time as the one that has performed at least one query in the previous month. Using this definition, Fig. 4 draws the time evolution of active users. We can see that the number of active users soared after the application was presented in a news conference, but then it fell steadily before stabilizing at around 500 active users per month. On the other hand, Fig. 5 shows the stability of the users in the sensor network. Notice that the drop out rate mainly affect new users, which means that the time users need to evaluate the utility of the application is short. This fact is important regarding the Validation Module, based on reputation, since the performance of this kind of measures improves for long term users.

5.2 Quantification Benefits of SSN

In this section, we provide some data retrieved from the application that demonstrate empirically the potential of Smart Sensor Networks.

One of the advantages of a SSN is the speed at which a sensor network can be deployed. In Table 1 we present the data deployment of *Incidències 2.0*. Notice that in less than 36 hours we were able to deploy more than 890 sensor nodes in the Barcelona metropolitan area without any economic cost[2].

On the other hand, SSN deployment allows a wide geographical spread following the patterns of population density. Figure 6 shows the geographical distribution of active users at the moment of writing this paper in the surroundings of Barcelona metropolitan area.

[2] Users were aware of the application through the mass media (after a news conference we did) plus the viral effect of social networks.

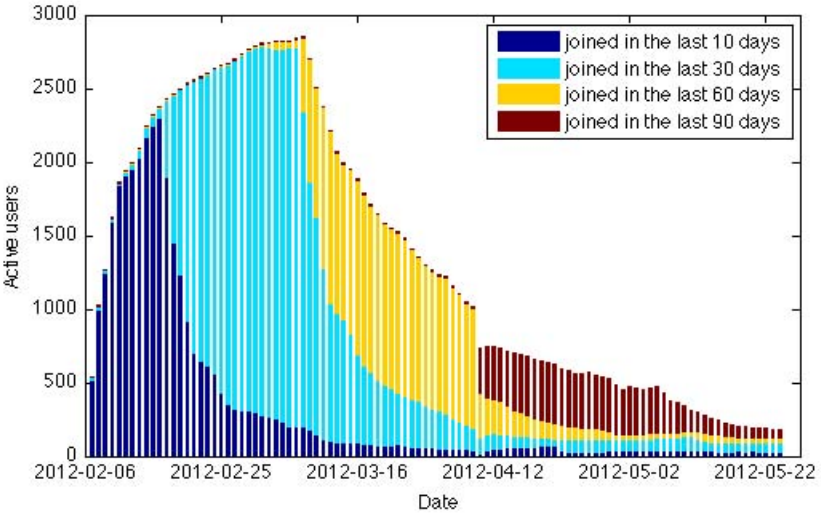

Fig. 5. Stability of the users in the sensor network

Table 1. Number of total accumulated installs by time intervals

	Days		
Hour	February the 5th	February the 6th	February the 7th
06:00	306	310	794
10:00	**306**	**326**	**890**
14:00	307	405	1001
18:00	307	571	1113
22:00	310	688	**1202**

Furthermore, sensor maintenance can be efficiently managed in a Smart Sensor Network. Traditional sensor networks entail a difficult process for software/firmware node's update process that is even harder in the case of wireless sensor networks due to the limitation of the transmission channel. However, SSN handle such process in a more simple way. For instance, 20 days after the massive deployment of our application, we add more features to allow users to sense new events. Such modification was deployed to the sensor nodes using the standard process of application upgrades defined in the Android market and Apple Store. Table 2 shows the update rate of the sensor nodes. It is to be mentioned that there were 982 active users at the time of the upgrade release. Notice that within less than a month more than 80% of the Android nodes and more than 69% of the iPhone nodes were updated and ready to use the new functionalities included for sensing new events. Even though the update process is straightforward, users can choose whether to upgrade the sensing application or not, or might not even be aware of the new release. Thereby, the latency of the update process might be higher compared to the latency of updating the nodes of a WSN.

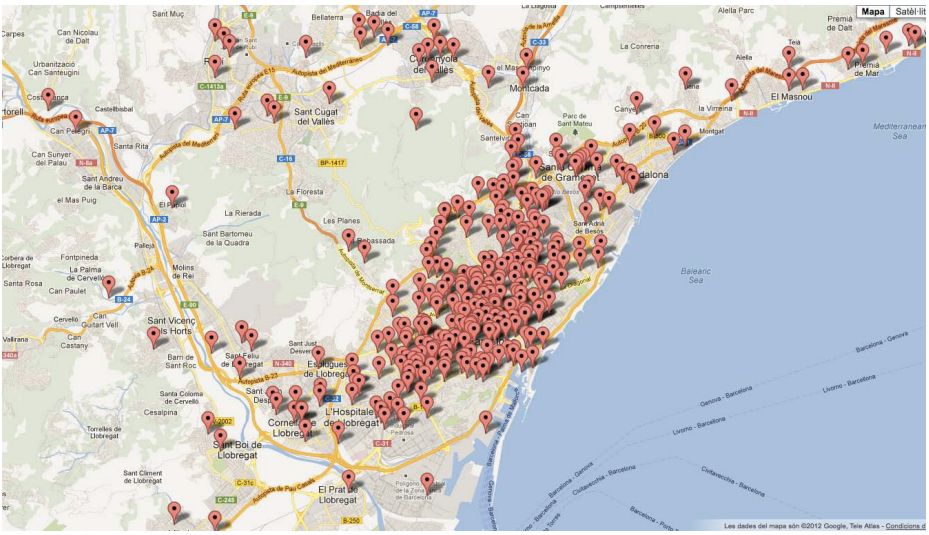

Fig. 6. Geographical deployment of sensor nodes (image showing aprox. 750 square kilometers around Barcelona)

Table 2. Percentage of updated sensor nodes by operating system

	Days			
OS	Feb. the 22nd	Feb. the 29th	Mar. the 14th	Mar. the 21st
Android	13,8%	47,5%	72,4%	80,2%
iPhone	0%	29,8%	64,7%	69,8%

6 Conclusion and Further Research

We believe that citizen-centring mobile sensing is becoming an important research area providing many interesting challenges from architectural to security and privacy specific. The wide spread and use of smartphones unfolds great potential to effectively map human-centring sensing tasks to end-user controlled smartphones. However, the architecture to support this kind of sensor networks bear little resemblance to the traditional wireless sensor network architecture discussed in the literature to date. In this paper, we have presented *Incidències 2.0*, a citizen-centric mobile sensing platform that allows individuals to report incidents in the railway transport services of the metropolitan area of Barcelona. The data obtained from the application shows us that it is possible to deploy a SSN in a very short period of time (almost 900 nodes in 36 hours) obtaining a wide geographic spread of nodes following the population geographic distribution. Nonetheless, this new vision of sensor networks raises complex privacy-related issues, that we intend to analyse and bring under discussion in further research.

References

1. Press Release. Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth, Gartner Newsroom (February 2012), http://www.gartner.com/it/page.jsp?id=1924314 (last access March 26, 2012)
2. Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: Proc. of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, pp. 53–62 (2006)
3. Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., Peterson, R.: People-centric urban sensing. In: Proc. of the 2nd Annual International Workshop on Wireless Internet, WICON 2006. ACM, New York (2006)
4. Eisenman, S., Campbell, A.: SkiScape sensing. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys 2006, pp. 401–402. ACM (2006)
5. Eisenman, S., Miluzzo, E., Lane, N., Peterson, R., Ahn, G.S., Campbell, A.: BikeNet: A mobile sensing system for cyclist experience mapping. ACM Trans. Sen. Netw. 6(1), 6:1–6:39 (2010)
6. Eunkyung, K., Luyan, C., Yu-Han, C., Maheswaran, R.: Dynamics of Social Interactions in a Network Game. In: 2011 IEEE Third International Conference on Social Computing (Socialcom), Privacy, Security, Risk and Trust, pp. 141–148 (October 2011)
7. Froehlich, J., Dillahunt, T., Klasnja, P., Mankoff, J., Consolvo, S., Harrison, B., Landay, J.: UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In: Proceedings of the 27th International Conference on Human Factors in Computing Systems, pp. 1043–1052. ACM (2009)
8. Gahran, A.: Reporting on the gulf oil spill from your cell phone (June 2010), http://articles.cnn.com/2010-06-11/tech/oil.spil.app_1_cell-phones-apps-geotagged?_s=PM:TECH (last access March 26, 2012)
9. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation system for online service provision. Decision Support Systems 43, 618–644 (2007)
10. Leijdekkers, P., Gay, V.: Personal heart monitoring and rehabilitation system using smart phones. In: International Conference on Mobile Business, ICMB 2006, p. 29 (June 2006)
11. Lysyanskaya, A., Triandopoulos, N.: Rationality and Adversarial Behavior in Multi-party Computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)
12. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Using mobile phones to determine transportation modes. ACM Trans. Sen. Netw. 6(2), 13:1–13:27 (2010)
13. Rose, G.: Mobile phones as traffic probes: practices, prospects and issues. IEEE Spectrum 38(1), 90–91 (2001)
14. Sayda, F.: Involving LBS users in data acquisition and update. In: Proceedings of the AGILE 2005, Conference on Geographic Information Science (2005)
15. Thompson, C., White, J., Dougherty, B., Schmidt, D.C.: Optimizing Mobile Application Performance with Model–Driven Engineering. In: Lee, S., Narasimhan, P. (eds.) SEUS 2009. LNCS, vol. 5860, pp. 36–46. Springer, Heidelberg (2009)