Mobile Context Provider for Social Networking

André C. Santos[1], João M.P. Cardoso[2], Diogo R. Ferreira[1], Pedro C. Diniz[1]

[1] IST – Technical University of Lisbon, Portugal
[2] Faculty of Engineering, University of Porto, Portugal
Corresponding author: `diogo.ferreira@ist.utl.pt`

Abstract. The ability to infer user context based on a mobile device together with a set of external sensors opens up the way to new context-aware services and applications. In this paper, we describe a mobile context provider that makes use of sensors available in a smartphone as well as sensors externally connected via bluetooth. We describe the system architecture from sensor data acquisition to feature extraction, context inference and the publication of context information to well-known social networking services such as *Twitter* and *Hi5*. In the current prototype, context inference is based on decision trees, but the middleware allows the integration of other inference engines. Experimental results suggest that the proposed solution is a promising approach to provide user context to both local and network-level services.

1 Introduction

The processing capabilities of mobile devices coupled with portable and wearable sensors provide the basis for new context-aware services and applications tailored to the user environment and its daily activities. To enable such kind of services, mobile devices must be able to clearly and accurately identify specific user contexts [1, 2]. For this purpose, mobile devices can be augmented with sensors that yield information such as position, lighting or sound, from which specific user contexts can be determined.

In this paper we present a prototype system consisting of a smartphone augmented with an array of sensors connected via *bluetooth*, and we describe the use of this system as a context provider for social networks. The system is an evolution of a previous prototype [3] and it is able to learn a set of given contexts and to identify the user context dynamically at run-time. The new prototype described in this paper is also able to publish context information to social networking services. The success of social networking services such as *Facebook*, *Hi5*, *LinkedIn* and *Twitter* and their increasing use via mobile devices suggest that there is an interest and potential in providing users with mechanisms to automatically keep up to date with their peers.

The paper is organized as follows. We begin with a description of related work in the area of sensing and context inference. Then we present the approach and the implemented system: its sensors, architecture, and operation through data acquisition, preprocessing, feature extraction, and context inference. Finally, we

describe how the system publishes user context to well-known social networking services, namely *Twitter* and *Hi5*.

2 Related Work

Context identification has been recognized as an enabling technology for proactive applications and context-aware computing [4, 5]. Early context-aware applications were predominantly based on user location defined as typical user places (e.g., "at home", "in a museum", "in a shopping center"). Projects such as GUIDE [6] and Cyberguide [7] addressed the use of information about location and situation to guide the user when visiting tourist attractions. Recent work has addressed techniques to identify a richer set of contexts or activities. These include simple user activities (e.g., "walking", "running", "standing"), environment characteristics (e.g., "cold", "warm"), or even emotional condition of the user (e.g., "happy", "sad", "nervous").

Usually, the identification of contexts is performed in several stages. Processing raw data from sensors may require a wide variety of techniques such as noise reduction, mean and variance calculation, time- and frequency-domain transformations, estimation of time series, and/or sensor fusion. Then context inference itself has been addressed using different techniques such as Kohonen Self-Organizing Maps (KSOMs) [8], k-Nearest Neighbor [9], Neural Networks [10], and Hidden Markov Models (HMMs) [11]. Some approaches also combine several of these techniques, as described in [12].

Regarding the inference of user activities such as "walking" or "running", there have been a myriad of approaches, ranging from simple processing steps and threshold operations [13, 14, 2] to the use of neural networks as a clustering algorithm [10]; or even using non-supervised time-series segmentation [15]. As an example, the work presented in [1] infers activities such as "walking", "running", "standing", and "sitting" with a single 3-axis accelerometer.
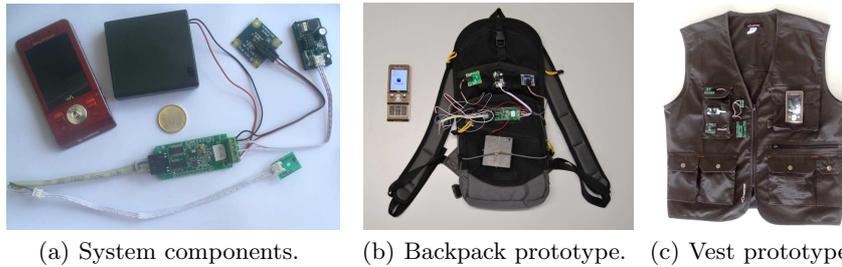
With the increasing popularity of social networking services, context information finds a new role in enabling interaction between members in a community. In [16], for example, the authors present the Meeting Room Assistant application, a hyper-local facilitator where context is used to enrich communication between participants in real-life meetings, in a similar way to what social networking services achieve on the web.

In the work presented in this paper, we extract signal features using techniques similar to those described in [2, 14]. For context inference we combine signal-processing and machine-learning techniques, using decision trees [17] to fuse features and to identify user activities. All data preprocessing and context inference is performed on the mobile device. The results can be published to a remote server in order to aggregate information from multiple users, thus allowing for more advanced and possibly non-local context inferences. Our work focuses in particular on the publication of context information to well-known social networking services.

3   System Prototype

The driving goal for this work is the ability to identify user context using a set of sensors connected to a mobile phone. The sensors should be relatively small in order to be embedded in clothes or personal items such as backpacks or purses. Sensors include accelerometers, light, sound, humidity, temperature and GPS sensors, and also *virtual sensors* to acquire information such as time of day and calendar events, which can be retrieved directly from the mobile device.

Figure 1(a) depicts the main components of the system prototype: the mobile device, a sensor-aggregating node, and a set of sensors. The black box contains the batteries (the 1-Euro coin is shown to provide an idea of scale). Figures 1(b) and 1(c) depict experimental setups where the components are embedded in a backpack and on a vest, respectively. The vest prototype is an evolution from the earlier backpack model. These experimental prototypes were used for testing purposes and therefore have deliberately unconcealed sensors in order to better evaluate sensitivity to the environment to ensure that the sensors experience approximately the same conditions as the user.

(a) System components.   (b) Backpack prototype.   (c) Vest prototype.
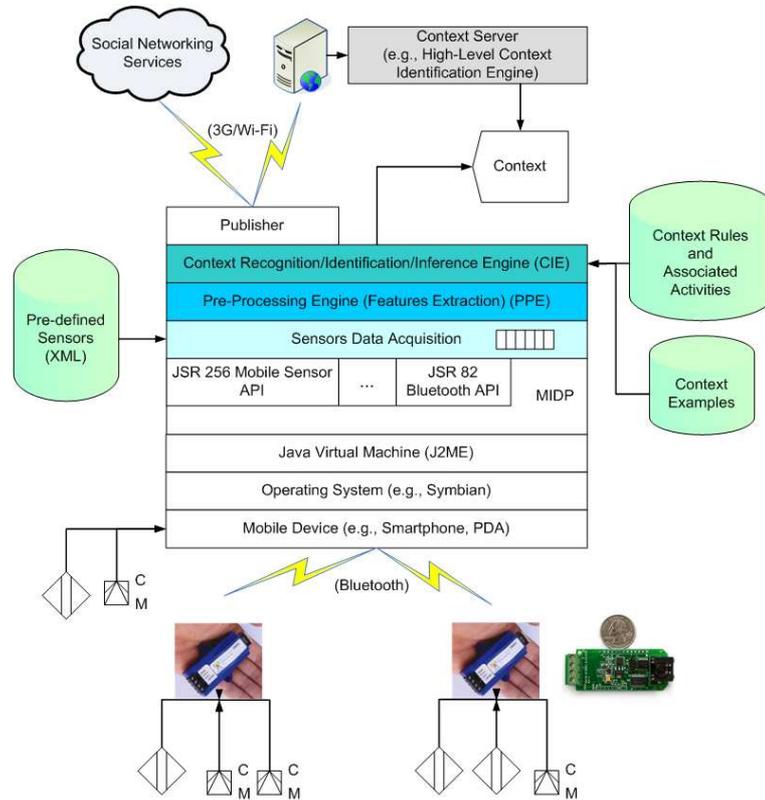
Fig. 1. System components and experimental prototypes.

Currently, the system is being used with either the *Sony Ericsson W910i* mobile phone or the *Nokia N95* mobile phone, but it can be easily deployed to other smartphones as well. There is a *BlueSentry* external sensor node that communicates with the smartphone via bluetooth to provide sensor readings, thus avoiding the need for physical connection between the two. With respect to sensors, the prototype includes sound, temperature, light, and humidity sensors, and a 3-axis accelerometer, all wired to the sensor node. In addition to these, there are three other sensors being used, namely the internal accelerometer of the smartphone, a virtual time sensor to provide the time of day from the system clock, and an additional external node which is a bluetooth GPS receiver.

3.1   System Architecture

The system makes use of supervised learning techniques to determine user context. During a training period, the system collects a sufficient number of manu-

ally classified examples in order to induce a decision tree that will be used for context identification. After this training phase the system operates autonomously and unobtrusively by automatically determining the present context from sensor readings. The overall system architecture is presented in Figure 2.

Fig. 2. System architecture with layer description and communication connections.

At the lowest level, sensors gather data from the environment and provide it as raw analog signals to the sensor node, which in turn converts them to digital form and transmits them to the smartphone via bluetooth. The application layer has been developed using the *J2ME* platform. The mobile phone runs a proprietary operating system which supports J2ME MIDlets. With the help of a Mobile Information Device Profile (MIDP), the application acquires raw sensor data from both the internal sensors and external sensor nodes, namely the BlueSentry aggregating node and the bluetooth GPS sensor.

The system operates according to four main stages: a sensor data acquisition stage; a preprocessing and feature extraction stage; a context inference stage; and a publication stage. Sensor data acquired from the available sensors are fed

to the preprocessing stage, which is responsible for extracting signal features to be used in the upper layers of the system architecture. The inference stage gathers these features and determines the present context according to a set of rules. These rules are available in the form of a decision tree that has been built during the training phase. Finally, the mobile device, upon having recognized a specific context, can provide that information to network-level services.

3.2 Application Layer

Figure 3 provides some screenshots of the application running on the smartphone. It presents a simple user interface, allowing different modes to be chosen from a list of available options (Figure 3(a)). It also includes the possibility of editing existing contexts (Figure 3(b)) and printing the decision tree for debugging purposes. Figure 3(c) presents an example of the initial configuration for the continuous training mode which acquires sensor data during a period of time when a certain context is active. In Figure 3(d) shows the different sensor readings and the identified context, as well as a confidence value calculated as the percentage of total records for the displayed context within a fixed-size buffer window. A suggestive icon, when available, is also presented to the user.

Fig. 3. The application in selection mode (a), in context-editing mode (b), in learning mode (c) and in operation mode (d).

3.3 Sensor Data Acquisition

Sensor data are acquired at a fixed rate. At regular intervals, the smartphone sends a request to the sensor node in order to retrieve data from the connected sensors. The sensor readings are buffered in the smartphone in order to allow a time-framed window preprocessing stage. It is worth noting that sensors may have different acquisition rates. The difference in sampling frequency may force the acquisition to run at the slowest rate, or at individual rates for each sensor. Currently, the system is using the same acquisition rate for all sensors, except for

the internal accelerometer of the smartphone, which is being sampled at twice the rate of other sensors.

3.4 Preprocessing and Feature Extraction

The preprocessing stage prepares the raw sensor data to be converted into a finite set of features or categories. While for some sensors the raw sensor value can be mapped directly to a category, other sensors such as the accelerometer need more elaborate preprocessing. Typically, preprocessing involves a set of techniques such as averaging, filtering or transforming values. Rather than using instant values, the system averages signals over a buffer window in order in order to minimize jitter and to provide a more accurate categorization. Table 1 presents the categories and their corresponding value range for each sensor.

Sensor	Category	Calculation	Sensor	Category	Calculation
sound	very silent	0% — 20%	temperature	very cold	$-50° — 0°$
	silent	20% — 40%		cold	$0° — 15°$
	moderate	40% — 60%		mild	$15° — 25°$
	loud	60% — 80%		hot	$25° — 30°$
	very loud	80% — 100%		very hot	$30° — 150°$
light	very dark	0 lx — 200 lx	time	dawn	0h — 5h
	dark	200 lx — 400 lx		morning	6h — 11h
	normal	400 lx — 600 lx		afternoon	12h — 17h
	bright	600 lx — 800 lx		night	18h — 23h
	very bright	800 lx — 1000 lx	movement accelerometer	not moving	low variance
GPS	indoor	no signal		moving	high variance
	outdoor	signal		moving fast	variance- and FFT-based
position accelerometer	lying	gravity xz-plane	humidity	low	0% — 30%
	standing	gravity y-axis		medium	30% — 70%
				high	70% — 100%

Table 1. Categorization of sensor values after acquisition.

For the movement accelerometer, which is internal to the smartphone, the system has more complex preprocessing. It computes the variance for each axis within a time-framed window which captures the last 16 samples. The three axis variances are compared to a threshold in order to identify the "moving" and "not moving" categories. When "moving" is detected, the system performs an FFT (Fast Fourier Transform) over the last 32 samples and adds up the amplitude of the harmonics for frequencies within the range from 0.5Hz to 2Hz. If the resulting value is greater than a specific threshold value then a category of "moving fast" is reported. Otherwise, the output will be reported simply as "moving".
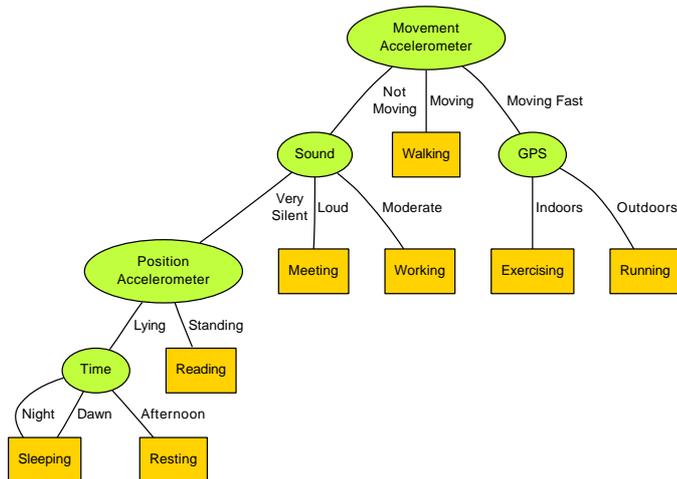
3.5 Context Inference

At any given moment, the collected set of sensor readings will bear some relationship to the current user activity. The purpose of context inference is to

discover this relationship so that when similar readings occur the device will recognize the same context.

The current prototype uses decision trees for context inference. They are fast to build and process, making them attractive for implementation on mobile devices. From a set of training examples it is possible to induce a decision tree in order to classify contexts according to sensor readings. The implementation is based on the ID3 algorithm [18], which uses information entropy to build the smallest tree that can correctly represent all branches.

Figure 4 depicts an example of an induced decision tree. The contexts depicted in this decision tree are an example of the type of activities that the system can be trained to recognize. The tree has paths from root to leaf nodes that effectively provide the rules for context identification based on sensor readings. For example, if the user is not moving and the sound level is moderate, then the context is "working". In this example, the movement accelerometer provides the most distinctive feature, followed by the sound and GPS sensors, and only then by the position accelerometer and the virtual time sensor.

Fig. 4. Example of an induced decision tree depicting several contexts and the sensor readings through which they can be identified.

The decision tree shown in Figure 4 can be induced from a set of training examples collected during a training period. One such example is, for instance: {sound = *moderate*; light = *very bright*; temperature = *mild*; humidity = *medium*; pos.acc. = *standing*; mov.acc. = *not moving*; time = *afternoon*; gps = *indoors*; context = *working*}. When the user sets a context and a training period as in Figure 3(c), the system collects sensor readings during that period and classifies the readings as training examples for the specified context. The training examples are then used to update the decision tree.

At runtime, the context identified via the decision tree is stored in a buffer window which gathers a finite number of records and returns the context that has been recorded more often within that time window. This avoids momentary conditions that would make the context change unexpectedly, and also provides a mechanism for the system to assign a confidence level to the context that has been inferred. Together with sensor data buffering, this provides another layer to mitigate errors due to faulty sensor readings or due to activities that are detected for only a brief moment within a different context.

3.6 Context Publication

The context can be published, upon user permission, to an external server. This has several advantages. First, it is possible to enable, disable or change the behavior of value-added services that depend on user context. Second, user context can be augmented with information available at the network level, such as traffic conditions or special services available at the user location. Third, the publication and availability of user context at the network level opens up the way to other applications such as social networking, remote monitoring, health assistance, etc. Context publication also provides network services with the ability to gather aggregated data on multiple users in order to study different user profiles.
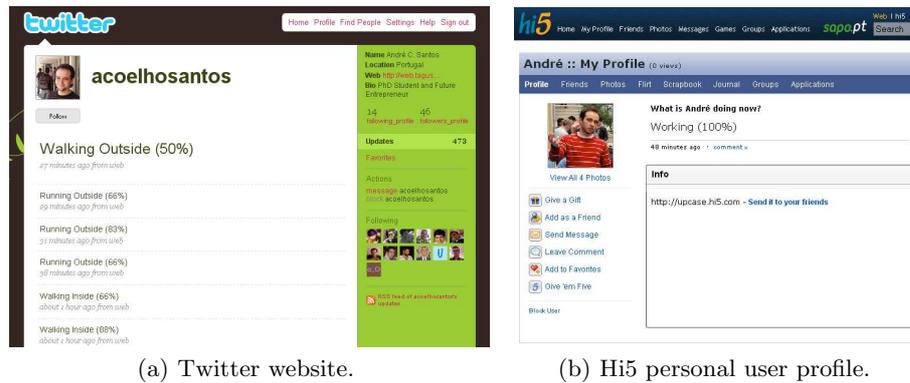
4 Application to Social Networking

Social networking has become increasingly popular with the rise of web-based communities in which users interact with their peers on a regular basis. Social networking services such as *Facebook*, *Hi5*, *LinkedIn* or *Twitter* nowadays bring together a large number of users. Most of these services provide public APIs with the means to access, configure and update the user profile, status, etc. This allows external systems to publish content to a user profile.

Twitter is a well-known social networking and micro-blogging service on which users can interact using short messages, no longer than 140 characters. We use the above system prototype to publish the current user context to a *Twitter* account. Periodically, the present context is submitted via HTTP through a Wi-Fi or GPRS/3G connection to the *Twitter* servers. The publication is accomplished using a REST API. Figure 5(a) presents a screenshot from the *Twitter* service, showing contexts published automatically by the system.

The same approach can also be used with other social networking services such as *Hi5*. This is one of the most successful web-based social networks with around 60 million users. The service allows users to create an online profile and populate it with information about interests, age and location, along with personal photo albums and music preferences, to be shared with friends.

Similar to *Twitter*, the *Hi5* service allows the user to post a status message to the network with information about the current activity the user is doing. Our system automates this task by publishing the user context periodically, so that the user profile is always up-to-date. For this purpose the system uses a Wi-Fi

(a) Twitter website.   (b) Hi5 personal user profile.

Fig. 5. User context publication to *Twitter* and *Hi5*.

or GPRS/3G connection, whichever is available at the moment, and makes a remote call using the official REST API. Figure 5(b) presents a screenshot from a *Hi5* profile page, where the status message is being published automatically by the system. Other possibilities include changing the user photo or changing the online/offline/away/busy state according to the present context, as well as tagging content with the current user context.

5 Conclusion

Being able to gather and publish information about user context can be very useful for social networking, as it enables real-time interaction between peers. In this paper we described a prototype system based on a set of general-purpose, inexpensive sensors connected to a smartphone via a bluetooth-enabled sensor node. The current prototype is implemented in a vest and executes all stages from sensing to context inference in the smartphone. It is now a fully operational context provider that operates in an unobtrusive manner after an initial training period, and that is able to publish the user context to well-known social networking services such as *Twitter* and *Hi5*. In ongoing work, we are evaluating the use of other context inference techniques, and we are also investigating the possibility of inferring emotional contexts such as happy, calm, unhappy, and hungry, in order to publish this information to social networks.

References

1. Kawahara, Y., Kurasawa, H., Morikawa, H.: Recognizing user context using mobile handsets with acceleration sensors. In: IEEE Intl. Conf. on Portable Information Devices (PORTABLE'07). (2007) 1–5

2. Welbourne, E., Lester, J., LaMarca, A., Borriello, G.: Mobile context inference using low-cost sensors. In: Location- and Context-Awareness. Volume 3479 of LNCS., Springer (2007) 254–263

3. Santos, A.C., Tarrataca, L., Cardoso, J.M.P., Ferreira, D.R., Diniz, P.C., Chainho, P.: Context inference for mobile applications in the UPCASE project. In: Proc. 2nd Intl. Conf. on Mobile Wireless Middleware, Operating Systems, and Applications (MOBILWARE 2009). Number 7 in LNICST, Springer (2009) 352–365

4. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is Key. Communications of the ACM **48**(3) (March 2005) 49–53

5. Hull, R., Neaves, P., Bedford-Roberts, J.: Towards situated computing. In: Proc. of the Intl. Conf. on Wearable Computers (ISWC'97). (1997) 146–153

6. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project. In: Proc. 6th Annual Intl. Conf. on Mobile Computing and Networking, ACM (2000) 20–31

7. Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: A Mobile Context-Aware Tour Guide. In: Proc. of the Intl. Conf. on Mobile Computing and Networking (MobiCom'96). (1996) 421–433

8. Laerhoven, K.V.: Combining the kohonen self-organizing map and k-means for on-line classification of sensor data. In: Artificial Neural Networks (ICANN 2001). (2001) 464–470

9. Laerhoven, K.V., Cakmakci, O.: What shall we teach our pants. In: Proc. of the Proc. Fourth Intl Symp. Wearable Computers (ISWC'00). (2000)

10. Randall, C., Muller, H.: Context awareness by analyzing accelerometer data. In: Proc. 4th Intl Symp. on Wearable Computers (ISWC'00). (Oct. 2000) 175–176

11. Skaff, S., Choset, H., Rizzi, A.: Context identification for efficient multiple-model state estimation. In: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). (Oct. 2007) 2435–2440

12. Krause, A., Smailagic, A., Siewiorek, D.: Context-aware mobile computing: Learning context-dependent personal preferences from a wearable sensor array. IEEE Trans. on Mobile Computing **5**(2) (Feb. 2006)

13. Healey, J., Logan, B.: Wearable wellness monitoring using ecg and accelerometer data. In: Proc. of the Ninth IEEE Intl. Symp. on Wearable Computers (ISWC'05), Washington, DC, IEEE Computer Society Press (October 2005) 220–221

14. Si, H., Kawahara, Y., H. Kurasawa, H.M., Aoyama, T.: A context-aware collaborative filtering algorithm for real world oriented content delivery service. In: Proc. of ubiPCMM. (2005)

15. Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J., Toivonen, H.: Time series segmentation for context recognition in mobile devices. In: Proc. of the 2001 IEEE Intl. Conf. on Data Mining (CDM'01), Washington, DC, IEEE Computer Society Press (Nov. 2001) 203–210

16. Joly, A., Maret, P., Daigremont, J.: Context-awareness, the missing block of social networking. International Journal of Computer Science and Applications **4**(2) (February 2009)

17. Quinlan, J.: Induction of Decision Trees. Machine Learning **1**(1) (March 1986) 81–106

18. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kauffman, San Francisco, Calif. (1993)