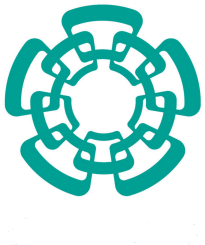

Research Center for Advanced Studies
from the National Polytechnic Institute

Information Technology Laboratory

Technical Report:

Smart Usage of Context Information for the Analysis, Design, and Generation of Power-Aware Policies for Mobile Sensing Apps

Rafael Pérez Torres, César Torres Huitzil and Hiram Galeana Zapién

Cinvestav Tamaulipas. Parque Científico y Tecnológico TecnoTam – Km. 5.5 Carretera Cd. Victoria-Soto La Marina. C.P. 87130 Cd. Victoria, Tamps.

LTI-TR-2015-07

Cd. Victoria, Tamaulipas, Mexico. December, 2015

Abstract:

This technical report describes the progress and advances developed for the thesis work titled *Smart Usage of Context Information for the Analysis, Design, and Generation of Power-Aware Policies for Mobile Sensing Apps* during the first year of the doctoral program.

In summary, the different efforts during the first year have been focused on producing a solid state of the art revision, as well as on defining the core elements that will power up the methodology for solving the main problem.

KEYWORDS: mobile sensing apps, energy consumption, context information, policy, smartphone

Corresponding author: Rafael Pérez Torres <rperez@tamps.cinvestav.mx>, César Torres Huitzil <ctorres@tamps.cinvestav.mx> and Hiram Galeana Zapién <hgaleana@tamps.cinvestav.mx>

Date of submission: December 5th, 2014

Rafael Pérez Torres, César Torres Huitzil, Hiram Galeana Zapién. Smart Usage of Context Information for the Analysis, Design, and Generation of Power-Aware Policies for Mobile Sensing Apps. CINVESTAV Tamaulipas. 2014 Nov. 36 pp. Technical Report No. LTI-TR-2014-07

Place and date of publication: Ciudad Victoria, Tamaulipas, MEXICO. December 16th, 2015

Contents

1 Introduction

This technical report describes the progress and advances developed for the thesis work titled *Smart Usage of Context Information for the Analysis, Design, and Generation of Power-Aware Policies for Mobile Sensing Apps* during the first year of the doctoral program.

In summary, the different efforts during the first year have been focused on producing a solid state of the art revision, as well as on defining the core elements that will power up the methodology for solving the main problem.

In order to ease the comprehension for the reader, this report has been structured as a self-contained document, briefly addressing the scientific context of the research, describing the hypothesis, problem statement and objectives. This technical report is structured as follows. Section 2 presents a brief review of the context of the research, recalling the characteristics of smartphone-based sensing applications and the problematic arised by the energy consumption when continuous access to sensors is needed. Section 3 is prepared as for describing the study performed on state-of-art techniques, covering a taxonomy of existing solutions, as well as a framework for their analysis. Section C is aimed at describing the advances produced on our proposed solution, including the insights for achieving learning from user and a general overview of the mechanisms for achieving sensor usage adaptations. Section D present the future work and specific activities for achieving the planned solution. Section E summarizes the scientific products prepared as a result of the first year of work. Finally, Section F present the conclusions of this report.

2 Research description

This section is aimed at giving an overall description of the context of the research, in order to ease the comprehension of the reader and producing a self-conteined document.

The popularity of smartphones is a remarkable instance of the adoption of technology by society. According to the Ericsson Mobility report [15] there were 3,400 million of smartphone subscriptions and 250 million of mobile PCs, tablets and mobile router subscriptions in the 2015. The presence of

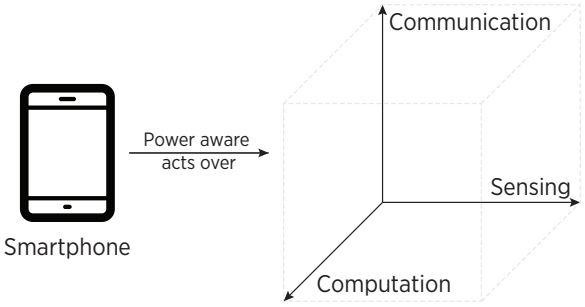

Figure 1: The communication, computation, and sensing features offered by smartphones represent the main dimensions where the functionality of these devices relies on.

smartphones is drastically changing our daily lives by providing an increasingly powerful set of tools as a result of significant advances in the field. The capabilities and platform complexity of smartphones are continuously improving, becoming truly system-on-chip (SoC) devices able to adapt their operation over three distinguishing dimensions, namely communication, sensing, and computation, as conceptually shown in Figure 1. The advances on these dimensions have enabled the *context-awareness* [1] paradigm in mobile and pervasive computing, opening the path for a myriad of applications and services. Particularly in the sensing dimension, modern smartphones include a set of low cost and powerful embedded sensors such as accelerometer, digital compass, gyroscope, GPS receiver, microphone, and camera, among others, providing them with rich capabilities for creating mobile sensing applications at different spatial and temporal scales [7, 21, 23] and with different levels of accuracy and energy costs. For this class of applications, the core requirements are the background and continuous sensors' data acquisition, as well as the associated on-device computations for extracting useful information for the user [23, 38]. As user interaction can be required for collecting data, two sensing paradigms can be implemented [23]. The *opportunistic paradigm* tries to determine the most appropriate moment for automatically collecting data from sensors without human participation at all. On the other hand, the *participatory paradigm* leverages user's abilities requiring his/her participation to describe data and choose the moment for collecting them. Given its autonomy, the opportunistic paradigm represents the preferable choice for continuous mobile sensing.

[1] For the sake of clarity, we stick to the definition of *context* proposed by Chen and Kotz [8], as the set of environmental states and settings that either determines the application's behavior or in which an application event occurs and is interesting to the user.

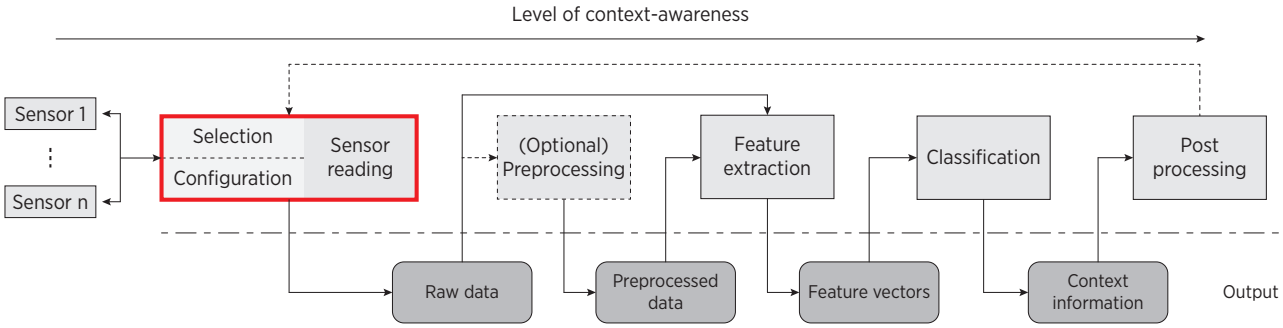

Figure 2: The different stages of mobile sensing applications produce output, shown in the lower gray rounded boxes, whose level of information is increasing at each stage. The post-processing stage might launch a new sensor reading chain as shown by the top dashed line.

2.1 Smartphone-based sensing characteristics

Regardless of the sensing paradigm, the internal operation of mobile sensing applications commonly involves a set of stages that describe the closed loop shown in Figure 2, consisting in: (a) *Sensor reading* that involves selecting, configuring, and requesting data to sensors, (b) An *optional preprocessing* for discarding uninteresting data like outliers or reducing noise, (c) *Feature extraction* for obtaining features, i.e., more computationally efficient representations of data, (d) *Classification* of extracted features into classes of special interest, for example human activity aspects that might feed machine learning strategies, and (e) *Post-processing* for offering feedback to user, launching network communication, triggering another sensing-classification chain with the outcome of classification acting as feedback information for a more precise operation of sensors, or for performing further processing. At each stage, the level of context-awareness of the mobile platform is increased, requiring specific algorithmic solutions to perform the above described functions. The design of such solutions is not straightforward, as there are a number of factors that impact the operation of each stage. This is the case of the dynamics of user's context which depends on changes produced in the environment, the existing constraints in mobile platforms, and the privacy concerns of personal data manipulation that, as a whole, compromise the sensor readings, classification and post-processing of data. Similarly, the implementation of these applications is hindered by priorities and additional requirements like the latency at which a context change is detected, the accuracy of the inferred context, and the implicit rational usage of energy resources.

Despite the fact there is a common understanding of how to deploy the aforementioned stages in

mobile sensing applications, the support for continuous sensing is still an open research problem due to the power demands of long-term sensing and the inherent resource-constrained characteristics of mobile devices. Although modern smartphones feature increasing computing power, memory and longer battery life, the requirements of complex mobile sensing applications represent a great challenge for mobile platforms, especially to the battery consumption, whose typical capacity is on the order of thousands of mAh, growing only 5-10% each year [16, 27]. This growing rate is not at the same pace than the advances in hardware and software components of mobile platforms, which in every new generation of mobile devices impose an ever growing energy demand.

At the same time, it is important to note that since their initial conception, mobile platforms described a layered architecture with a clear focus on managing the heavy interaction with user. because of this, the support for background running processes and for exploiting the power management capabilities existing at hardware level of sensors was discarded, leaving out the power-efficiency of the sensing dimension [37]. As a consequence, efficient implementations of power-aware mechanisms that do not jeopardize the accuracy requirements of mobile sensing applications still remain an open challenge, which is augmented by the computational and power constraints of mobile devices, as well as the long-term and near real-time requirements of these applications.

2.2 Hypothesis

As described in previous section, long-time operation of mobile sensing applications remains as an open challenge, due to the inherent trade-off between the energy consumed when accessing to sensors and the accuracy of the activity being tracked. In simpler terms, the more accuracy obtained when accessing sensors with high frequency, the higher energy consumption. Nonetheless, this research work considers that the dynamics of context information inferred from sensors data is of remarkable value for adapting sensory operations and producing power savings. Thus, the hypothesis pursued in this work states that *intelligent policies produced through context information built from sensors data can be employed to reduce the energy consumption in a mobile device when performing continuous sensor readings*.

In a deeper description, a smart policy is a special rule that defines how sensors should be accessed in order to reduce the energy consumption and achieve mobile app requirements. This research work aims to employ data coming from GPS and inertial sensors in order to obtain context information in terms of

mobility patterns, useful to reduce the energy consumption generated by LBS's [2].

2.3 Problem statement

In order to prove the aforementioned hypothesis, this research work identifies the existence of two interrelated problems, which are described as follows. The first one is depicted as a *pattern identification problem* and is focused on detecting changes in the mobility pattern of user from sensors data. Such pattern represents the context information that is helpful to add *smartness* to the policies that adapt sensory operations.

Formally: Given a set $V = \{v_1, v_2, \ldots, v_n\}$ of data values read from sensor S in the time interval $T = [t_1, t_2]$, find the behavior pattern $Pattern_S$ that represents the activity of user.

$$PatternIdentifier(V) \longrightarrow Pattern_S \in Patterns \tag{1}$$

Where $Patterns$ is a set of patterns that represent an interesting state in the user mobility data, namely $\{no\_movement, walking, running, vehicle\_transportation\}$.

On the other hand, the *policy generation problem* is related to the selection and configuration of proper sensors for keeping the user location tracking, while at the same time reducing energy consumption. For this process, it is also important to consider the level of the smartphone's battery and the accuracy requested by the mobile application.

Formally: Given the detected pattern $Pattern_S$ in data from sensor S, parameters for assigning weight to energy eh, and physical constraints status pc of a mobile device, find a policy to adapt the duty cycle of sensors.

$$PolicyGeneration(Pattern_S, eh, pc) \longrightarrow DutyCycle_S \tag{2}$$

2.4 Objectives

Main objective

To reduce energy consumption in the mobile sensing apps, which perform continuous sensor readings, through self-adapting power-aware policies generated from context information obtained from sensors data.

[2] LBS, Location Based Service.

Particular objectives

- To identify mobility patterns from context information obtained from an inertial sensor (accelerometer) and location providers (GPS, WPS).
- To generate policies for a self-adapting sensors' usage from identified mobility patterns, accuracy and energy requirements of mobile application, and status of mobile device's constraints.
- To ease the development of mobile sensing applications that require user location tracking, i.e., LBS, isolating the complexity of sensors' access and the associated efficient energy management.

2.5 Contributions

- A mechanism for detecting mobility patterns from the data read by sensors of mobile devices (especifically GPS and accelerometer).
- A mechanism for generating policies for accessing sensors. Such mechanism employs application requirements (energy and precision hints), level of mobile device constraints, and user's context information (using the pattern detected by previous mechanism). The produced policies will allow to perform a smarter usage of smartphone's sensing infrastructure in continuous sensor readings, reducing the energy consumption.
- A middleware that implements the previously described mechanisms, easing the development of mobile sensing applications.

3 State-of-art works analysis

The content of this section describes the results of the analysis of state-of-art related works. A taxonomy of these solutions is presented, and the pure hardware and hardware-software approaches are briefly described. Finally, the pure software approach is presented at detail, highlighting its advantages over other approaches, and including a framework for decomposing and studying solutions under this category.

Power-aware sensing refers to a set of techniques and methodologies aimed at continuously monitoring sensor data over long periods of time under the computing, storage, and power constraints of typical mobile

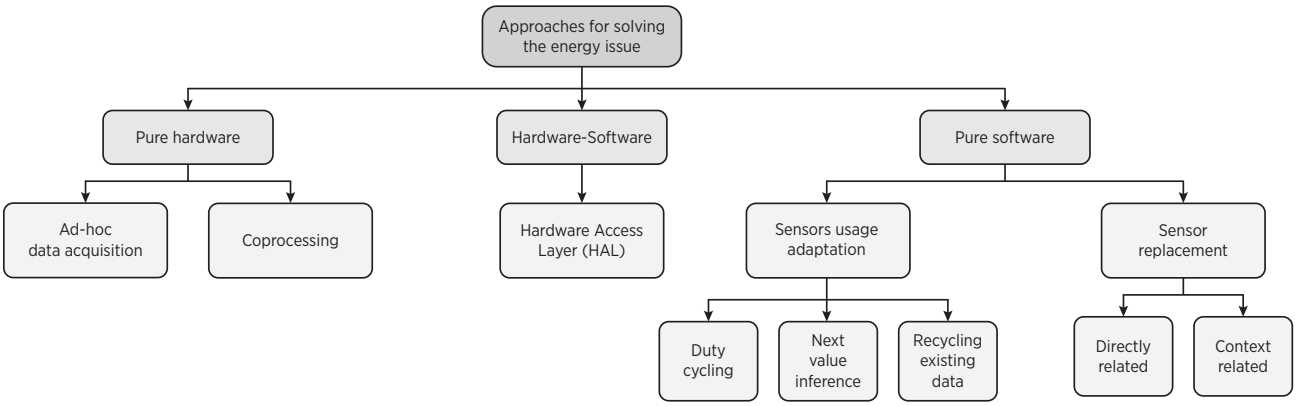

Figure 3: The taxonomy of solutions for power-aware smartphone sensing is divided into three categories: pure hardware, hardware-software and pure software.

devices. Roughly speaking, the power-aware sensing is mostly enclosed into the sensor reading stage of mobile sensing applications, abstracting the complexity of sensors management, and delivering collected data while considering the power constraints of these devices. It is important to remark that this goal is frequently achieved by executing the rest of the stages internally, in order to identify context information that is useful for feedback purposes and a smarter usage of sensors.

A revision of relevant works in literature reveals three complementary approaches at different level of abstraction of the smartphone device: pure hardware, hardware-software, and pure software approaches. Figure 3 presents a taxonomy of solutions for power-aware smartphone sensing, with the pure hardware, hardware-software and pure software approaches as the main classes, briefly described in the next paragraphs. It has been identified that the support for these policies at system level of a mobile platform facilitates the analysis, control, and cross-layer coordination and information sharing for reducing the power consumption, which is the reason why power-aware sensing mechanisms should be supported throughout all layers of the mobile platform [39]. Figure 4 illustrates the distribution of the different approaches across the layers of a mobile platform, highlighting the increasing support in the higher layers for reacting accordingly to the dynamics of context information through more complex and robust machine learning mechanisms.

It is important to highlight that although the presented taxonomy allows to classify the power-aware smartphone-based sensing techniques, the relevance of this problem makes that in practice many of the solutions combine different strategies pursuing an integral and better power management. This is a suggestion of the complexity of the problem itself and an evidence of the evolution and adoption of these

Year	Name	Mobile platform	Description	Approach
[11] 2008	*The Mobile Sensing Platform*	Intel iMote with custom hardware	User motion	Pure hardware
[37] 2011	*LittleRock*	Not specified	User motion	Pure hardware
[5] 2013	*Apple iPhone 5s*	Apple iPhone 5s and newer	User motion	Pure hardware
[47] 2010	Not given	Android G1 (Android 1.5)	User location	Hardware-Software
[38] 2012	Not given	Samsung SGH-i917 (Windows Phone 7.5)	Generic sensors access	Hardware-Software
[12] 2009	*EnLoc*	Nokia N95 (Symbian 60)	User location	Pure software
[1] 2009	*SenseLess*	Nokia N95 (Symbian 60)	User location	Pure software
[43] 2009	*EEMSS*	Nokia N95 (Symbian 60)	User location, user activity	Pure software
[22] 2009	*EnTracked*	Nokia N95 (Symbian 60)	User location	Pure software
[28] 2009	*iLoc*	Simulation	User location	Pure software
[33] 2010	*RAPS*	Nokia N95 (Symbian 60)	User location	Pure software
[19] 2010	*SensLoc*	Simulation	User location	Pure software
[35] 2010	*G-Sense*	Simulation	User location, user activity	Pure software
[24] 2010	*A-Loc*	Simulation and Android G1	User location	Pure software
[26] 2010	*Jigsaw*	Nokia N95, Apple iPhone	User location, user activity	Pure software
[10] 2011	*SmartDC*	HTC Desire (Android 2.1)	User location	Pure software
[34] 2011	*CAPS*	Nexus One (Android 1.5-2.2)	User location	Pure software
[40] 2012	Not given	Simulation	User activity	Pure software
[36] 2012	Not given	Samsung Galaxy S (Android 2.3)	User location	Pure software
[46] 2013	*SensTrack*	Google Nexus S (Android 4.1.4)	User location	Pure software
[30] 2013	*Not given*	Custom Samsung Galaxy Nexus, and Samsung Galaxy S4 (Android)	User location	Pure software
[9] 2014	*FreeTrack*	Smartphone not specified (Android 2.2)	User location	Pure software
[29] 2014	Not given	Samsung Galaxy III (Android 4.1.4)	User location	Pure software
[44] 2014	Not given	Blackberry Storm II	User activity	Pure software
[14] 2014	Not given	HTC MyTouch 3G, Google Nexus One, and others (Android)	User location, user activity	Pure software
[3, 31] 2014	Not given	Google Nexus One	User activity	Pure software
[18] 2015	Not given	Simulation (of energy harvesting device)	User activity	Pure software

Table 1: A set of produced works aimed at solving the power consumption issue in smartphone-based mobile sensing. Most of the solutions follow a pure software approach, given its flexibility and adaptive features.

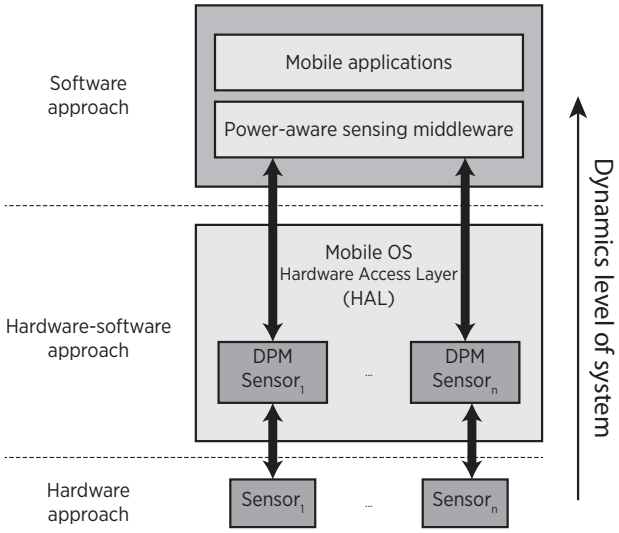

Figure 4: The approaches distributed across the mobile platform stack are more adaptive to system's dynamics on the higher layers.

complementary techniques. Furthermore, ideas proved to be working efficiently in software soon or later are implemented directly in hardware to boost efficiency even further.

Table 1 summarizes a set of works found in literature aimed at solving the power consumption issue in mobile sensing applications in the context of the presented taxonomy. As it can be noted, the most common strategy followed in the literature is the pure software approach, due to its flexibility and fine-tuning features, which will be described in next sections of this paper. Because of the heterogeneity of target features, main purpose, and mobile platforms where the mechanisms proposed in each work were implemented, it is not straightforward to perform a fair comparison among them [32, 42]. For example, the solutions may target a different set of sensors or the techniques for analyzing context information produce outputs with different semantic meaning. Also there is no a clear consensus for measuring the energy benefits of proposed solutions; in this regard two main strategies have been followed. The first and most common strategy is to perform a comparison in terms of mW between energy consumed by a solution and the mobile platform or a similar approach. However, even inside this strategy the implementations are not homogeneous and the proposed solutions employ different methods for measuring energy consumption in mobile devices, namely highly-invasive, moderately-invasive, and non-invasive techniques, as described in [2]. The second strategy performs the comparison in terms of battery life (e.g., time to drain) because, according to [20], several physical characteristics of batteries impact on their discharging cycle. The most

Name	Variant	Sensors involved
LittleRock [37]	Ad-hoc data acquisition	Accelerometer, compass, gyroscope
Apple iPhone 5s [5]	Coprocessing	Accelerometer, compass, gyroscope
The Mobile Sensing Platform [11]	Coprocessing	Accelerometer, barometer, compass, humidity, light, microphone, temperature

Table 2: Representative works under the pure hardware approach. More context-aware features are being deployed directly into hardware components of mobile platforms.

notorious aspect is that the actual available energy is reduced when the battery discharges, a behavior that is ignored in the energy consumption models included in most of the reviewed solutions.

3.1 Pure hardware approach

The fundamental idea behind the pure hardware approach is the selection of power-aware hardware elements for providing physical data to upper layers of the mobile platform, as well as the definition of mechanisms to adapt the hardware input parameters, like DVFS. These mechanisms allow the creation of control points for manipulating hardware usage, which are known as power modes [6, 25, 39]. The mobile platform keeps the list of power modes fixed and instruct sensors to work isolated from others. In this way, the hardware components obey a static behavior defined only through power modes, whose control points are exported to upper layers of the mobile platform.

The set of works under this approach typically involve the design of specific hardware to exploit the aforementioned features. This design implies the isolation of the components associated with the measurement of physical signals inside a new unit with a dedicated low power processor (LPP). Since the mobile OS isolates the access to these hardware components, mobile applications remain agnostic about the new hardware unit and can consume its power-aware sensing services without modifying their inner logic. The basic behavior of this unit requires sensors to report their readings to the embedded LPP, which is capable of executing light preprocessing operations like noise filtering. While the hardware unit performs the reading and filtering stages, the rest of the smartphone hardware platform is able to reach its sleep mode, producing power savings. Table 2 shows some of the most representative works under the pure hardware approach. Depending on the ability to process data and discover contextual information autonomously, the pure hardware approach can be implemented in two different ways, namely the *ad-hoc data acquisition* and the *co-processing* variants, as shown in Figure 3.

Reference	Produced HAL	Sensors involved
[38]	API for selecting processor for execution of arbitrary tasks	Accelerometer, microphone
[47]	Automatic substitution, suppression, piggybacking, and adaptation of location providers	GPS, wireless interface

Table 3: Representative works under the hardware-software approach. The produced hardware interfaces improve the overall efficiency of the mobile platform.

The **ad-hoc data acquisition** variant is restricted to serve the different mobile application requests for raw sensor data, discarding the analysis and extraction of higher level information.

Besides the raw data delivery, the **co-processing** variant adds support to hardware layers for light data processing and detection of higher level information, producing context-awareness at hardware level with low power consumption. In this sense, the sensing facilities of the hardware platform are transformed into smart sensors [17]. The embedded LPP of the sensing unit must be powerful enough to perform such analysis.

Smartphone manufacturers have begun to integrate into the device's hardware full suites of sensing capabilities. For instance, Apple has implemented a platform re-design of the *iPhone* smartphone, which starting at *5s* version features an isolated sensing unit able to process sensor data and deliver fitness information like the amount of steps and distance covered by user, as well as the type of activity performed, namely, stationary, walking, running, automotive, cycling, and unknown [5].

Regardless of the variant, a common aspect taken into account in the design of pure hardware approach solutions is the power consumption - usage generality trade-off that emerges when hardware designers are unable to foresee the needs of any future mobile application and implement power-aware mechanisms to support them directly in the circuits. Because of this limitation, the effort of designers is focused on implementing general features that can be employed by many mobile sensing applications, instead of very specific features meaningful only for a few of them.

3.2 Hardware-software approach

The hardware-software approach is aimed at defining system-wide policies for deciding when to turn sensors on and off, or when to switch to a different power mode of a given hardware component. In this level, the hardware-software approach is aware of the existence of different sensors and coordinates the basic operations

and interactions between them in a DPM fashion, being able to abstract fine grained parameters into a coarser grained set, isolating the hardware usage for mobile applications and elements in upper platform layers. Unlike the pure hardware approach, the hardware-software approach offers a basic understanding of global activities being performed by the mobile device, and also an implicit observance of dynamic changes in the workload of hardware components for deciding when to adapt hardware operation. Because of the coupled interaction with hardware components of the platform, hardware-software approach solutions are produced as HAL's of the mobile OS or low level middlewares that build an interface that abstracts the access to sensors and isolates their complex management mechanisms. Table 3 presents a few instances of works that follow this approach this approach.

3.3 Pure software approach

Thanks to the context information obtained from sensor data, the pure software approach can bring activity awareness to the mobile platform and make informed decisions for fully dynamic power-aware sensors management as illustrated in Figure 4, where the pure software approach is at the top of the hierarchy. Typically, pure software approach solutions are powered up by a set of classifiers fed with sensors data that individually identify basic aspects or produce abstractions about user's activity. Such user activity aspects are the fundamental elements for defining sensor management policies. In this way, the pure software approach is able to make the mobile platform aware about user activity and even adapting accordingly to changes on its profile. In power-aware computing terms, the pure software approach can be understood as *spending power to save power* [39]. In this sense, there is an unavoidable amount of energy that is spent while performing computational processes for preprocessing, classifying and discovering context information from sensor data; however, substantial benefits might be also obtained [45]. Firstly, this approach achieves reduction in power consumption by means of understanding user activity. Secondly, the information inferred from sensors data allows mobile devices to improve their awareness about high level user activities.

Solutions following this approach can only observe the details of the hardware platform exposed by the hardware access or lower layers of the mobile OS. A practical interpretation can be expressed as follows: these lower layers know how to turn circuits on and off, but are unable to define when; whereas the higher software layers are flexible and can dynamically adapt to changes in user context, knowing when adapt the sensors operation, but delegating how to do it to the lower layers. Typically, pure software approach

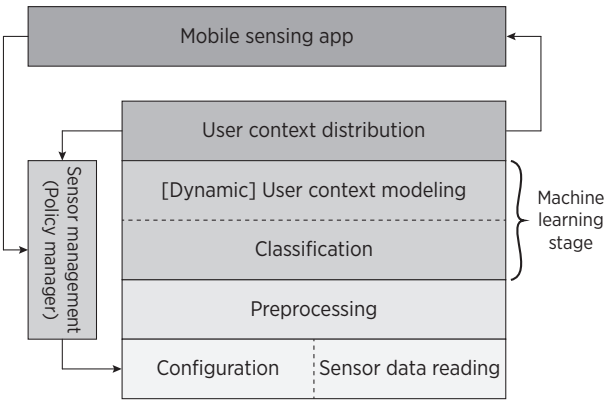

Figure 5: A generic structure of a middleware following a pure software approach defines explicit modules for selecting and configuring sensor access, and classification and machine learning stages.

solutions are implemented through a layered middleware with classification and machine learning modules embedded on it, as depicted in Figure 5. This middleware is placed between running mobile applications and the sensing and communication layers of the mobile platform, abstracting its sensing and communication facilities [44].

Table 4 shows a set of works following the pure software approach. With the aim of providing a detailed analysis of relevant aspects of this approach, the rest of this section is structured as follows. In first term, the categories or variants of this approach are enumerated and described. Next, relevant characteristics found in these solutions are described. Finally, a framework for their decomposition and analysis is presented. BRIEFLY WORK ON IT!

3.3.1 Categories of the pure software approach

Depending on the type of decision made for modifying hardware access, two variants of the software approach can be identified, namely the *Sensors usage adaptation* and *Sensor replacement*.

3.3.1.1 Sensors usage adaptation It is based on the analysis of historical data in a time window for detecting patterns or events in the user behavior. Once a pattern is detected it can be employed in the next three paths for adapting sensors usage and reduce power consumption:

- **Duty cycling (DC)**: The frequency for accessing sensors is reduced, introducing sleep periods in hardware, and hence achieving power savings. However, this also produces the accuracy - power

Name	Variants	Machine learning technique	Sensors involved	Complexity	OL	OO	US
G-Sense [35]	Sensor adaptation (DC)	SDR	GPS	low			
Not given [36]	Sensor adaptation (DC)	SDR	GPS	low			
SenseLess [1]	Sensor adaptation (DC), Sensor replacement (CR, DR)	SDR	WPS, GPS, ACC	low			
SensTrack [46]	Sensor adaptation (DC), Sensor replacement (CR, DR)	SDR	ACC, orientation sensor, GPS, WPS	low			
Not given [29]	Sensor adaptation (DC, VI), Sensor replacement (CR)	SDR	ACC, magnetic field sensor, GPS	low			
EnLoc [12]	Sensor adaptation (DC, VI), Sensor replacement (DR)	SDR, Mobility Tree	WPS, GPS, cellular ID	medium		✓	
EnTracked [22]	Sensor adaptation (DC), Sensor replacement (CR)	SDR	ACC, GPS	medium		✓	
Not given [3, 31]	—	Ameva algorithm	ACC	medium			
Not given [30]	Sensor replacement (CR)	DT	Temperature, humidity, pressure	medium			
Not given [40]	Sensor adaptation (DC)	DT	ACC	medium			
Not given [18]	Sensor replacement (CR)	KNN	Model of ACC-based harvesting device	medium			
SensLoc [19]	Sensor adaptation (DC, RD), Sensor replacement (CR)	SDR	Wi-Fi fingerprinting, GPS, ACC	medium	✓		
CAPS [34]	Sensor adaptation (DC, RD), Sensor replacement (CR)	SDR	GPS, cellular ID	medium	✓		
RAPS [33]	Sensor adaptation (DC, RD), Sensor replacement (CR, DR)	SDR	WPS, GPS, ACC, Bluetooth, cellular ID	medium	✓		
A-Loc [24]	Sensor adaptation (DC, RD), Sensor replacement (CR, DR)	HMM, Bayesian estimation framework	GPS, WPS, Bluetooth, cellular ID	medium	✓	✓	
SmartDC [10]	Sensor adaptation (DC, RD), Sensor replacement (CR, DR)	HMM and LZ predictor	GPS, WPS, Wi-Fi and cellular ID fingerprinting	medium	✓	✓	
Jigsaw [26]	Sensor adaptation (DC), Sensor replacement (CR)	Microphone: NB with Gaussian Mixture Model (GMM). ACC: DT. GPS: MDP.	ACC, Microphone, GPS	high		✓	✓
Not given [14]	Sensor adaptation (DC)	Several. KNN and NN selected as best.	ACC, GPS, WPS, cellular ID, light, device data, mobile app requirements	high			✓
EEMSS [43]	Sensor adaptation (DC), Sensor replacement (CR, DR)	GPS and ACC: SDR. Microphone: SSCH algorithm.	ACC, microphone, GPS	high			✓
iLoc [28]	Sensor adaptation (RD), Sensor replacement (CR)	HMM	Wi-Fi & GSM fingerprinting	high	✓		✓
Not given [44]	Sensor adaptation (DC, RD)	HMM	ACC	high	✓		✓
FreeTrack [9]	Sensor adaptation (DC, RD), Sensor replacement (CR, DR)	HMM	GPS, Wi-Fi, cellular ID, battery status	high	✓		✓

Table 4: Important works proposed following the pure software approach. Many of these works combine different software strategies for boosting their energy performance. (OL: Online Learning from user data, OO: Optimization Oriented solution, US: User State context insight, ACC: Accelerometer).

consumption trade-off, since the continuous data acquisition is interrupted and mobile applications would work with such data gaps.

- **Next value inference (VI)**: Obtains an inference of the next value to be delivered by sensors, avoiding an actual access to them, and hence reducing power consumption; however, this introduces uncertainty to the system which may be not acceptable for certain applications. Typically, the inference is produced by probabilistic measures focused on low level dependencies observed in short-time windows of raw sensor data.

- **Recycling existing data (RD)**: Identifies common circumstances or events in high level context information that happened previously, reusing data from such events. In practice, this is feasible since people describe a consistent behavior, which can be identified by looking at patterns in their activities in long time windows [10,34]. Since actual access to sensors is avoided, it may also introduce uncertainty to the system if the probability of accurately identifying the event is low.

3.3.1.2 Sensor replacement It permits a replacement of the sensor currently employed for collecting data with another one that consumes less power and has immediate data availability. Replacing a sensor may involve the definition of a mapping function between the data type of the selected sensor and the replaced one. Such mapping function is needed when the output of sensors is not compatible; for instance, accelerometer data can used for inferring distance, but only after proper processing (integrating for speed and position). Depending on whether such mapping is needed, two specializations can be found:

- **Directly related replacement (DR):** The sensor is replaced with a lower power consumer one that delivers the same data type and guarantees the application's accuracy requirements, without the need of a mapping procedure.

- **Context related replacement (CR):** The type of the data returned by the selected sensor is not as expected by the application, implying a mandatory procedure for translating it to the requested data type.

A noteworthy aspect about pure software approach solutions is that typically once their power efficiency is proved in tests and real life implementations, they might be implemented in the hardware or lower layers of the mobile OS. For instance, modern inertial sensor devices incorporate an internal buffer that allows

data queuing until it is full or the phone is woken up; also, recent GPS receivers are able to implement directly in hardware *geo-fencing* features [41], allowing to wake up the smartphone when it enters in the geo-fenced area. Both functionalities were initially available in the Android platform via software but now are supported at hardware level [4, 13]. Also, lower layers of newer versions of Android allow the *piggy-backing* of close in time GPS requests, reusing the same GPS fix (a location point obtained through the GPS infrastructure) for concurrent mobile applications [47]. This transference of platform functionalities from software to hardware is always subject to the aforementioned trade-off between the generality of features and power saving benefits: a very generic mechanism can be suitable for any mobile application but will not obtain the largest power savings; on the other hand, the best power savings are obtained by the most specific and tuned solutions that may result useless for the rest of mobile application scenarios.

3.3.2 Relevant characteristic of the software approach - Analysis of power-aware solutions under the pure software approach

The openness and flexibility of the software stack of mobile devices allows to implement a wide variety of strategies and mechanisms that can dynamically adapt to changes detected in context information. In addition to such diversity of strategies, there is also heterogeneity in the scenarios (like LBS, HAR, crowd sensing) and objectives pursued by the different pure software solutions, which hinders their review and makes it hard to produce a definitive framework for their analysis. As an example of the aforementioned heterogeneity, the reader can consider works that duty cycle sensors usage. In some proposed solutions, the context information employed in the policies for adapting duty cycle is simple, even raw data with simple heuristics; while on the other hand, some solutions build and process a more complex model of context information before performing a decision with more robust classifiers. In this way, we identify that the granularity (or the level) of context information represents the pivotal feature on which the analysis and description of works can be performed.

The granularity of context information can be mostly characterized by the combination of the next elements: (a) the data type of the input directed to classifiers, (b) the classifier itself, and (c) the length of the time window employed to sense data. Recall that although solutions can follow unconstrained combinations over these elements for inferring context information, at the end the mechanisms for adapting sensors operation will incur into one or more variants of the pure software approach, described in

Section 3.3.1. Also, it is worth noting that among the different features described by solutions, we have identified the following attributes as helpful to analyze and understand core features of their design and operation. First, solutions can incorporate mechanisms for online learning (OL) from context information, enabling predictive features thanks to observance over long-time windows of sensory data. Also, works can follow and optimization orientation (OO) approach in order to formalize, for instance, a sensor usage adaptation (replacement) mechanism and guarantee a given target in terms of minimum energy consumed and/or the error in activity tracking. Finally, the solutions might employ an enriched version of the context information, known as user state (US) for allowing the device to become fully activity-aware and ease its adaptation over the sensing dimension.

Thus, with the granularity of context information insight and the aforementioned attributes in mind, we present the analysis and discussion of the pure software approach works listed in Table 4, highlighting their strengths and improvements, as well as their impact to establish a trend or new opportunities for research.

Bibliography

[1] Fehmi Ben Abdesslem, Andrew Phillips, and Tristan Henderson. Less is more: energy-efficient mobile sensing with senseless. In *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds.*, pages 61–62, 2009.

[2] David Perez Abreu and Maria E. Villapol. Measuring the energy consumption of communication interfaces on smartphones using a moderately-invasive technique. In *2012 Global Information Infrastructure and Networking Symposium, GIIS 2012*, pages 1–6, 2012.

[3] M. a. Álvarez De La Concepción, L. M. Soria Morillo, L. Gonzalez-Abril, and J. a. Ortega Ramírez. Discrete techniques applied to low-energy mobile human activity recognition. A new approach. *Expert Systems with Applications*, 41:6138–6146, 2014.

[4] Android Open Source Project Android. Batching, 2013.

[5] Apple. Core Motion Framework Reference, 2013.

[6] Luca Benini, Alessandro Bogliolo, and Giovanni De Micheli. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):299–316, 2000.

[7] Andrew Campbell and Tanzeem Choudhury. From smart to cognitive phones. *IEEE Pervasive Computing*, 11:7–11, 2012.

[8] Guanling Chen and David Kotz. A Survey of Context-Aware Mobile Computing Research. Technical report, 2000.

[9] Yohan Chon, Yungeun Kim, Hyojeong Shin, and Hojung Cha. Adaptive duty cycling for place-centric mobility monitoring using zero-cost information in smartphone. *IEEE Transactions on Mobile Computing*, 13(8):1694–1706, 2014.

[10] Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11*, page 82, 2011.

[11] Tanzeem Choudhury, Gaetano Borriello, Sunny Consolvo, Dirk Haehnel, Beverly Harrison, Bruce Hemingway, Jeffrey Hightower, Predrag Klasnja, Karl Koscher, Anthony LaMarca, James a. Landay, Louis LeGrand, Jonathan Lester, Ali Rahimi, Adam Rea, and Danny Wyatt. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, 7:32–41, 2008.

[12] Ionut Constandache, Shravan Gaonkar, Matt Sayler, Romit Roy Choudhury, and Landon Cox. EnLoc: Energy-efficient localization for mobile phones. In *Proceedings - IEEE INFOCOM*, number 4, pages 2716–2720, 2009.

[13] Google Developers. Geofence, 2014.

[14] Brad K. Donohoo, Chris Ohlsen, Sudeep Pasricha, Yi Xiang, and Charles Anderson. Context-aware energy enhancements for smart mobile devices. *IEEE Transactions on Mobile Computing*, 13(8):1720–1732, 2014.

[15] Ericsson. Ericsson Mobility Report. Technical Report November, 2015.

[16] Eric C. Evarts. Lithium batteries: To the limits of lithium. *Nature*, 526(7575):S93–S95, oct 2015.

[17] Stéphane Gervais-Ducouret. Next smart sensors generation. *SAS 2011 - IEEE Sensors Applications Symposium, Proceedings*, pages 193–196, 2011.

[18] Sara Khalifa, Mahbub Hassan, and Aruna Seneviratne. Pervasive self-powered human activity recognition without the accelerometer. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 79–86. IEEE, mar 2015.

[19] Donnie H Kim, Younghun Kim, Deborah Estrin, and Mani B Srivastava. SensLoc: Sensing Everyday Places and Paths Using Less Energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 43–56, New York, NY, USA, 2010. ACM.

[20] Minyong Kim, Young Geun Kim, Sung Woo Chung, and Cheol Hong Kim. Measuring Variance between Smartphone Energy Consumption and Battery Life. *Computer*, 47(7):59–65, 2014.

[21] Mikkel Kjaergaard. Location-based services on mobile phones: Minimizing power consumption. *IEEE Pervasive Computing*, 11:67–73, 2012.

[22] Mikkel Baun Kjaergaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjær. EnTracked : Energy-Efficient Robust Position Tracking for Mobile Devices. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 221–234, 2009.

[23] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(September):140–150, 2010.

[24] Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao. Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, page 285, 2010.

[25] J.R. Lorch and A.J. Smith. Software strategies for portable computer energy management. *IEEE Personal Communications*, 5(3):60–73, jun 1998.

[26] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The Jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*, page 71, 2010.

[27] Xiao Ma, Yong Cui, and Ivan Stojmenovic. Energy efficiency on location based applications in mobile cloud computing: A survey. In *Procedia Computer Science*, volume 10, pages 577–584, 2012.

[28] Yiming Ma, Rich Hankins, and David Racz. iLoc: a framework for incremental location-state acquisition and prediction based on mobile sensors. In *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, page 1367, New York, New York, USA, 2009. ACM Press.

[29] Yemao Man and Edith C H Ngai. Energy-efficient automatic location-triggered applications on smartphones. *Computer Communications*, 50:29–40, 2014.

[30] Sînziana Mazilu, Ulf Blanke, Alberto Calatroni, and Gerhard Tröster. Low-power ambient sensing in smartphones for continuous semantic localization. In *Lecture Notes in Computer Science (including*

subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8309 LNCS, pages 166–181, Ireland, 2013. Springer.

[31] Luis Morillo, Luis Gonzalez-Abril, Juan Ramirez, and Miguel de la Concepcion. Low Energy Physical Activity Recognition System on Smartphones. *Sensors*, 15(3):5163–5196, 2015.

[32] Steve Neely, Graeme Stevenson, Christian Kray, Ingrid Mulder, Kay Connelly, and Katie a. Siek. Evaluating pervasive and ubiquitous systems. *IEEE Pervasive Computing*, 7(3):85–88, 2008.

[33] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, volume 223-224, page 299, New York, New York, USA, 2010. ACM Press.

[34] Jeongyeup Paek, Kyu-Han Kim, Jatinder P. Singh, and Ramesh Govindan. Energy-efficient positioning for smartphones using Cell-ID sequence matching. In *Proceedings of the 9th international conference on Mobile systems, applications, and services - MobiSys '11*, page 293, New York, New York, USA, 2011. ACM Press.

[35] Alfredo J. Perez, Miguel a. Labrador, and Sean J. Barbeau. G-Sense: A scalable architecture for global sensing and monitoring. *IEEE Network*, 24(August):57–64, 2010.

[36] Rafael Perez-Torres and Cesar Torres-Huitzil. A power-aware middleware for location & context aware mobile apps with cloud computing interaction. In *Proceedings of the 2012 World Congress on Information and Communication Technologies, WICT 2012*, pages 691–696, 2012.

[37] Bodhi Priyantha, Dimitrios Lymberopoulos, and Jie Liu. LittleRock: Enabling energy-efficient continuous sensing on mobile phones. *IEEE Pervasive Computing*, 10:12–15, 2011.

[38] Mr Ra, Bodhi Priyantha, Aman Kansal, and Jie Liu. Improving Energy Efficiency of Personal Sensing Applications with Heterogeneous Multi-Processors. In *The 14th International Conference on Ubiquitous Computing*, pages 1–10, 2012.

[39] Parthasarathy Ranganathan. Recipe for efficiency. *Communications of the ACM*, 53(4):60, 2010.

[40] Vijay Srinivasan and Thomas Phan. An accurate two-tier classifier for efficient duty-cycling of smartphone activity recognition systems. In *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones - PhoneSense '12*, pages 1–5, 2012.

[41] Miguel Torroja, Steve Malkos, and Christophe Verne. Putting the (ultra-low) Power in GeoFence, 2013.

[42] Narseo Vallina-Rodriguez and Jon Crowcroft. Energy management techniques in modern mobile handsets. *IEEE Communications Surveys and Tutorials*, 15(1):179–198, 2013.

[43] Yi Wang, Jialiu Lin, Murali Annavaram, Quinn a Jacobson, Jason Hong, Bhaskar Krishnamachari, and Norman Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 179–192, 2009.

[44] Ozgur Yurur, Miguel Labrador, and Wilfrido Moreno. Adaptive and energy efficient context representation framework in mobile sensing. *IEEE Transactions on Mobile Computing*, 13(8):1681–1693, 2014.

[45] Ozgur Yurur, Chi Harold Liu, Zhengguo Sheng, Victor Leung, Wilfrido Moreno, and Kin Leung. Context-Awareness for Mobile Sensing: A Survey and Future Directions. *IEEE Communications Surveys & Tutorials*, (c):1–1, 2014.

[46] Lei Zhang, Jiangchuan Liu, Hongbo Jiang, and Yong Guan. SensTrack: Energy-efficient location tracking with smartphone sensors. *IEEE Sensors Journal*, 13(10):3775–3784, 2013.

[47] Zhenyun Zhuang, Kyu-Han Kh Kim, and Jp Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, page 315, 2010.