dWatch: a Personal Wrist Watch for Smart Environments

Dario Bonino, Fulvio Corno, Luigi De Russis*

Politecnico di Torino, Dipartimento di Automatica ed Informatica - Corso Duca degli Abruzzi 24 - 10129 Torino, Italy

Abstract

Intelligent environments, such as smart homes or domotic systems, have the potential to support people in many of their ordinary activities, by allowing complex control strategies for managing various capabilities of a house or a building: lights, doors, temperature, power and energy, music, etc. Such environments, typically, provide these control strategies by means of computers, touch screen panels, mobile phones, tablets, or In-House Displays. An unobtrusive and typically wearable device, like a bracelet or a wrist watch, that lets users perform various operations in their homes and to receive notifications from the environment, could strengthen the interaction with such systems, in particular for those people not accustomed to computer systems (e.g., elderly) or in contexts where users are not in front of a screen. Moreover, such wearable devices can reduce the technological gap introduced in the environment by home automation systems, thus permitting a higher level of acceptance in the daily activities and improving the interaction between the environment and its inhabitants. In this paper, we introduce the dWatch, a personal wearable notification and control device, integrated in an intelligent platform for domotic systems, designed to optimize the way people use the environment, and built as a wrist watch so that it is easily accessible, worn by people on a regular basis and unobtrusive.

Keywords: wearable computing, wrist watch, domotics, home automation, smart environments

1. Introduction

Homes and buildings with sensing, actuation, and networked devices have been anticipated for a long time. In such environments, users easily perform tasks involving different sets of devices and systems, like closing the shutters and turning the lights on at the sunset, without the need for advanced configuration or programming skills. For example, imagine a home with controllable lights, windows, doors and HVAC system. It should be easy to set up this environment to open a door with a movement of an arm, to reach a suitable in-door temperature based on the preferences of the present inhabitants or to promptly and personally inform the users about some unexpected events happened in the home.

Contemporary domotic systems (or home automation systems) try to ease the interaction between different home appliances and users by means of fixed, in-home, touch panels or with software applications for computers, tablets or smartphones. Such devices, however, have some known limitations:

- they are multi-purpose devices, and they could be busy by performing various activities or controlled by another home inhabitant (e.g., for gaming, web browsing, etc.);
- they typically lack environmental sensors, such as temperature sensors;
- even if some of them (e.g., tablets and smartphones) are portable, they are not always carried around, especially in the home [1];

*Corresponding author

Email addresses: `dario.bonino@polito.it` (Dario Bonino), `fulvio.corno@polito.it` (Fulvio Corno), `luigi.derussis@polito.it` (Luigi De Russis)

- they need to be picked up, opened or turned on before they can be accessed and fully used.

Moreover, there are situations where is not possible, secure or suitable to use one of these devices, for example under the shower, with wet hands, or when such devices are too distant to be immediately reached.

Wearable computing may provide a feasible solution to overcome many of the mentioned limitations, by reducing the technological gap introduced in the environment by home automation systems as much as possible, by enhancing the invisibility of domotics systems and by permitting a high level of acceptance in the daily activities of the users. While many wearable sensors have been developed, few are actually worn by people on a regular basis, due also to their size, position on the body, or since they are wired in cloths [2, 3]. A wrist watch is well known to be an attractive form factor for a wearable computer. It has the advantage of always being with you (even under the shower, if it is waterproof), and it can be instantly viewed with the flick of the wrist. Moreover, the form factor of a wrist watch makes it highly available and unobtrusive to users. In fact, a wrist watch is a very attractive for four main reasons [1]:

1. a large fraction of the population is already accustomed to wearing this type of watches;
2. watches are less likely to be misplaced compared to devices such as phones;
3. watches are more accessible than other devices one may carry;
4. a wirst watch is ideally located for body sensors [4] and as a wearable display [5].

The main drawbacks of using a wrist watch as a wearable platform are the usually short battery life, mainly due to the power absorbed by the wireless communication, and the limitations in visualizing information, due to the small display size. For wearable interfaces a long lasting battery duration is an important requirement [6].

In this paper, we introduce the dWatch, an off-the-shelf personal wearable notification, sensing and control device, integrated with an intelligent platform for smart environments. Internally, the dWatch includes a main board with a microprocessor, standard sensors, wireless communication, and a firmware/software that can be possibly replaced or enhanced by advanced users to customize the watch functionality for various applications. Its sensors (temperature and accelerometer, at least) enable a smarter usage of the environment, e.g., by allowing lighting systems to have optimized behaviors based on the user preferences, or by letting appliances to react to users' presence.

The dWatch is integrated in a smart domestic environment using an intelligent platform, i.e., a domotic gateway, that offers high manageability and flexibility by interfacing many different commercial domotic plants. In such an environment, the dWatch lets the user control home appliances, easily regulate the indoor temperature, play an alarm in case of emergency or as a reminder, receive notifications about customizable events and so on.

The remainder of the paper is organized as follows: Section 2 relates the proposed approach to the state of the art, highlighting commonalities and differences, and introduces some programmable watches available on the market. Section 3 presents some different use cases to exemplify the adopted solution, while Section 4 briefly describes the general logical architecture of an implemented system where the dWatch is included and explicits the dWatch requirements. Section 5 reports the complete description of the developed watch. In the end, Section 6 concludes the paper and depicts future works.

2. Related Works

Several "smart" wrist watches have been developed during the past 12 years. The first example of a smart watch dates back to 2000 with the IBM Linux Watch [1]. In its original form, the Linux Watch was a PDA on the wrist with no sensors; later revisions added accelerometers and audio sensors. It also acted as an alert notification device with wireless connectivity. It had a monochrome OLED screen and, as the name suggests, it ran Linux. This watch, however, did not communicate with any home automation system and had a power consumption too high for day long operations.

Another example of smart wrist watch is the one developed and sold by Microsoft during the period 2004-2008 by using the Smart Personal Object Technology (SPOT) [7]. The SPOT wrist watches, the first application of such technologies, used MSN Direct network services, delivered across the United States and Canada and based on FM radio broadcast signals. Microsoft watches were not research products but commercial watches, providing information such as weather, news, stocks, calendars, etc. on their LCD monochrome display. The entire MSN Direct service has run until recently, January 1st, 2012, when it was totally discontinued. These wrist watches were quite expensive (approximatively 200 $) and they did not communicate with any home automation system.

An interesting wrist watch is the eWatch [4], jointly developed by Carnegie Mellon University (USA) and Technische Universität München (Germany). The eWatch is a wearable sensing, notification and computing platform, with bluetooth communication capability to provide a wireless link to a cellular phone or a computer. The eWatch senses light, motion, audio and temperature, providing visual, audio and tactile notification. It has a battery that lasts multiple days, a monochrome LCD screen and it could be used for location and activity recognition. The eWatch has some interesting features, such as activity recognition, but it is not a commercial product and it is not able to interact with any home automation system.

Finally, a recent gesture-based wrist watch is the prototype developed by Research In Motion Limited [8]. Such a watch makes possible to acquire information from a companion mobile device through simple eyes-free gestures, with an haptic feedback. It is equipped with custom-made piezoelectric actuators, a force sensor and it is able of wireless communications through the bluetooth protocol. Such a watch, however, does not have any environmental sensor and does not communicate with any home automation system.

Interesting commercial products that could be good candidates for the dWatch hardware are the eZ430-Chronos, the "i'm watch" and the WIMM ONE. The eZ430-Chronos[1] watch is an off-the-shelf, low cost (approximately 50 $) programmable wrist watch produced by Texas Instruments. It may be used as a reference platform for watch systems, as a personal display for personal area networks, as a wireless sensor node for remote data collection, or simply as a watch. It features a LCD display and provides an integrated pressure sensor and an accelerometer. The eZ430-Chronos is water-resistant, offers temperature and battery voltage measurement and is complete with a USB-based wireless interface to a computer. Its firmware is totally customizable. Several applications have been developed for this tool, such as a wireless door lock[2] or its integration with the Z-Wave gateway Vera[3]. In the former case, however, the interaction with the door is made by using custom components, not directly available on the market, whilst in the second case the interaction with the automated house is much simpler than the one proposed in this paper and it only works with Z-Wave devices. The solution we introduce in this paper, instead, provides the interoperability of the dWatch with different commercial devices and protocols.

The "i'm watch"[4] and the WIMM ONE[5] appeared on the market in 2011. Both are Android-based platform. The former has a 1.54" touch screen display, a rechargeable battery, a bluetooth connection and is conceived as an extension of a smartphone. In some versions, it is provided with a magnetometer and an accelerometer. However, the "i'm watch" is expensive (its price ranges from 329 $ up to 2299 $, according to the chosen version and services), its battery lasts only up to 24 hours, with the wireless connection enabled, it does not have any environment sensor and does not communicate with any home automation system. The WIMM ONE is similar to the "i'm watch": it is an Android-based smart watch equipped with a capacitive touch screen, an accelerometer and a magnetometer, and audible and tactile alerts. It provides Wi-Fi and bluetooth connections to a computer or a smartphone, and it is splash resistant. It was born as a multifunctional platform, that can be used as a wrist watch but also as a necklet. Like the i'm watch, the WIMM ONE has a rechargeable battery (that lasts up to 36 hours), does not communicate with any home automation system, and does not have any environment sensor. It is quite expensive (approximatively 200 $) and currently it is only available as a Developer Preview.

3. Scenarios

In this Section we introduce three different scenarios, involving one or more people living in a home equipped with off-the-shelf home automation components. Scenarios describe some relevant features of the dWatch, by incrementally exploring its wide range of applications. The general architecture supporting these scenarios will be discussed in Section 4.

3.1. Home management with a single watch

Sam is a college student that lives in a domotic flat near the University. He is able to control all the appliances and the devices of his home by means of his computer, a small in-wall panel and some "traditional" buttons. However, in

[1] `http://processors.wiki.ti.com/index.php/EZ430-Chronos`
[2] `https://ziyan.info/2010/01/secure-wireless-door-lock/`
[3] `http://micasaverde.com/`
[4] `http://www.imwatch.it`
[5] `http://www.wimm.com`

some circumstances, he uses his dWatch. The dWatch, by default, shows him the time and the date but, by pressing one of the buttons on the watch, he may enter the main menu and perform different functions. For example, when he feels hot, he selects the "temperature" menu voice, sees the room temperature, and lowers the temperature in the room by pressing the down-arrow button on the watch. Moreover, Sam has set a shortcut on his dWatch: he configured two buttons of the watch to raise and lower the shutter of his bedroom, without entering the menu, but immediately in the default view (i.e., the one showing time and date).

Since Sam never reminds his deadlines, he uses the dWatch as a notification center. Sam sets a reminder on some of his deadlines and appointments, triggered by time and/or by his position. For example, if Sam has to buy some milk, the dWatch is able to remind him by playing an alarm and showing on his display "milk" when he is leaving the apartment. If the house needs to send some (unexpected) messages to the dWatch, for various purposes, they can be handled by the dWatch: if the fire system turns on in the kitchen, the watch plays an alarm and shows "fire - kitchen"; or if an anomalous electric power consumption is registered, the dWatch shows an appropriate message; etc.

Even the interaction with the appliances and the devices in the home is enhanced by the presence of the dWatch. In fact, the dWatch provides gesture-based interaction to manage different types of devices. For example, Sam can lower a shutter by moving his arm top-down, or open a door by horizontally moving his arm away from him, etc. To select the proper device to interact with, Sam can choose the device in a list shown by the dWatch, or he can simply approach the device to control.

Sam may enable, disable or modify some of these messages and behaviors by using a visual user interface in his computer or tablet, that lets the user easily set-up what message is sent when an event occurs, from which device or appliance, and to which watch (if more than a dWatch is present) [9].

3.2. Home management with multiple watches

The Janeway family is composed by four people: Jim and Kathryn, the parents, Tom, the teenager son, and Anne, a 8 years old girl. The family lives in a independent house, equipped with home automation systems. Kathryne is a designer and has her studio in the house. All the family members have a dWatch that acts as described in the previous scenario. Since more than a dWatch is present in the home, and their owners may have different "roles" and "privileges", each dWatch performs differently. For example, Tom's and Anne's dWatches cannot control any devices in their mother's studio, whilst their watches have some shortcuts to easily interact with devices of their bedroom. Similarly, the dWatch notifies Anne and Tom each time they forget their bedroom lights turned on.

In a family context, the dWatch also acts as a simple pager: if Anne needs her mother, she can play an alarm on the dWatch of Kathryne. Kathryne's dWatch will consequently play an alarm and show the message "Anne", since the caller is her daughter Anne. Moreover, if Kathryne is not at home (or her dWatch is not reachable), the system is able to forward the message to the adult-owned dWatch nearest to Anne, if any, or send the alert via SMS to Kathryne's phone.

3.3. Simple building management

Dave is a building manager, that lives in and manages an apartment block. Such a building is totally equipped with domotic systems, inside the flats, in the common areas (e.g., the entrance hall) and in the centralized plants (e.g., the HVAC system). People living in the building are equipped with a dWatch and, in their apartments, they may perform all the activities shown in the previous scenarios. Dave, as the building manager, has some more privileges and tasks.

When Dave is in the building, he receives a notification on his dWatch when something is going wrong: a lamp in the entrance hall that stopped working, or the HVAC system that went out-of-order, etc. Such a notification can be automatically generated by the automated system or sent by another joint owner. If one of these notifications is critical, the system delivers it to Dave, even if he is not in the building: i.e., the notification could be sent by SMS or e-mail.

Dave is also able to change the settings of centralized plants, like the HVAC system, by using the dWatch buttons or the gesture-based interaction, if he is close to the central heating system. Moreover, he is able to transfer some (or all) these tasks to another joint owner.

4. Logical architecture

The systems depicted in the previous scenarios consist, essentially, of three logical parts:

- some commercial home automation systems and components, even of different vendors and technologies (e.g., lamps, power outlets, shutter actuators, door actuators, etc.);
- a multi-protocol intelligent residential gateway, such as the Dog gateway [10], with suitable programming and configuration interfaces;
- the dWatch.

The dWatch is a personal wearable notification, sensing and control device that must obey to the following requirements: it must include some standard sensors (temperature and accelerometer, at least), localization and wireless communication, a good battery life, a display (possibly a touch-screen), some buttons, and a firmware/software that can be optionally replaced or enhanced by advanced users to customize the watch functionality for various applications. In particular, the dWatch should lets users control and set various domotic devices and appliances, through its menu, or by using advanced feature (e.g., gesture-based interaction); it receives and sends notifications and reminders from the smart environment, and from other dWatches; and it offers different functionalities according to the privileges of its owner, and lets the user transfer them to other dWatches.

To integrate the dWatch in a smart environment, we chose the Dog gateway as the intelligent platform. The Dog gateway [10] is a domotic gateway that implements a high-level semantic device modeling strategy (based on the DogOnt ontology [11]) and a pluggable driver architecture for supporting incremental and seamless integration of different home automation technologies. OSGi [12] is the gateway founding technology, supporting an extensible and modular architecture, based on downloadable applications known as bundles. For the purpose of this paper, Dog is important for:

- its ability to integrate different home automation technologies and let them interoperate by using suitable drivers;
- its ability to support the advanced functionalities of the watch by modeling them in DogOnt;
- the availability of bundles, such as DogRules, enabling the creation and the execution of "intelligent" tasks (e.g., cross-plant interoperability); or DogPower, able to split the metered aggregated power consumption values into device level estimated measures.

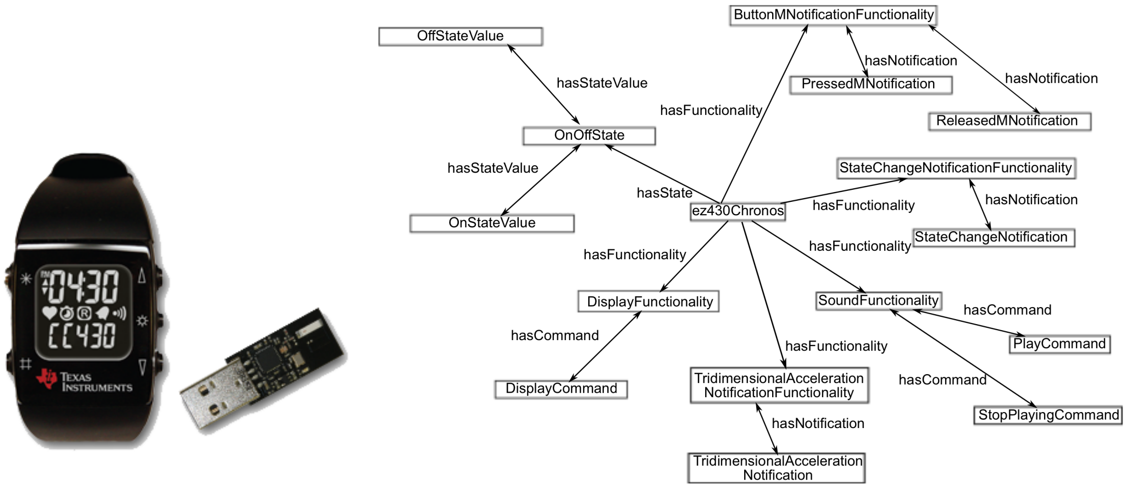
Dog communicates with the dWatch through a wireless connection, managed by a suitable Dog driver that implements the dWatch protocol.

5. dWatch design

Our first implementation of the dWatch interface device is based on the off-the-shelf eZ430-Chronos watch (shown in Figure 1(a)[6]), a highly integrated, wearable wireless development system that comes in a sports watch, developed by Texas Instrument. The watch features a 96 segment LCD display and five buttons, also providing an integrated pressure sensor and a 3-axis accelerometer. It is 30 meters water-resistant, offers temperature and battery voltage measurement and is able to communicate with a computer using two different wireless protocols, SimpliciTI or BlueRobin.

We chose this wrist watch for five reasons:

1. the eZ430-Chronos is a low-cost (approx. 50 $), off-the-shelf wireless product, distributed worldwide;
2. the eZ430-Chronos has a battery that lasts much longer than the other smart watches reported in the end of Section 2 (i.e., several months versus few days);
3. the watch comes with all the tools (for programming and configuration purposes) needed for working with it;
4. it has environmental sensors on board (e.g., temperature and pressure sensors);
5. it is a "zero-configuration" tool, since it is easily customizable by using the application it comes with.



(a) The eZ430-Chronos watch (b) The eZ430-Chronos modeled in DogOnt

Figure 1: The TI eZ430-Chronos watch and its modeling

The eZ430-Chronos supports all the requirements presented in Section 4, providing environmental sensors, wireless connection with a computer (i.e., the Dog gateway) through a USB dongle, a customizable firmware and a good battery life (several months, according to the Texas Instrument specification).

To turn the eZ430-Chronos watch into the dWatch device, we modeled the capabilities of the watch in DogOnt (shown in Figure 1(b)), and wrote a device driver for letting Dog communicate with the watch by means of an extension of the SimpliciTI protocol.

A custom firmware, written in the C language and based on the OpenChronos firmware[7], has been developed and loaded on the watch. The main menu proposed by the custom firmware is shown in Figure 2.

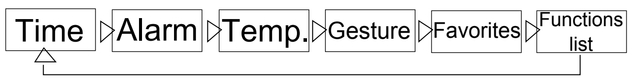


Figure 2: The dWatch Main Menu

The first menu item (*Time*) is the default view of the watch: it shows time and date. By pressing the “star” (*) button (the top-left button), it is possible to navigate through the menu, while by pressing and holding the “hash” (#) button (bottom-left) is possible to configure the options of the shown menu (i.e., to set time and date for “Time” menu). The second menu item (*Alarm*) has three functions: it sets sound alarms on the watch, acting like a “normal” alarm; it offers the possibility to send a reminder to another watch, if available; it shows the previous reminders and alarms, thus letting the user re-enable one of them. New text reminders must be created on a computer or on another device connected with the domotic plant, for example, by connecting Dog with the Google Calendar API and setting some reminders on the free Google service.

The third menu item (*Temperature*) shows the environment temperature and lets the user raise or lower it, by controlling the set-point of the HVAC system, while the fourth menu item (*Gesture*) enables/disables the gesture-based control of home devices. Table 1 reports a set of candidate gestures we are evaluating against existing user studies.

The fifth menu item (*Favorites*) shows the favorite devices, i.e., the home devices that the user plans to use or uses more frequently. Each device in this menu can be controlled, no matter the distance of the user from the device

[6]Picture taken from the eZ430-Chronos Wiki
[7]`https://github.com/poelzi/OpenChronos`

Gesture	Action	Closest devices or appliances
Moving the arm down-up	Raise	Shutters, Dimmer Lamps
Moving the arm up-down	Lower	Shutters, Dimmer Lamps
Moving the arm right-left	Open	Doors, Windows
Moving the arm left-right	Close	Doors, Windows
Draw a “O” in the air with the arm	Turn on	Lamps, Mains Power Outlets
Draw a “X” in the air with the arm	Turn off	Lamps, Mains Power Outlets

Table 1: First set of gestures supported by the dWatch

itself. Finally, the sixth menu item (*Function lists*) shows the privileges and the full list of functionalities the watch is enabled for, by also permitting to transfer some privileges to selected, nearby dWatches.

5.1. Dog integration

A new device driver has been included in Dog, supporting the modified SimpliciTI protocol used by the watch. The driver is written in Java, according to the OSGi guidelines. It opens a serial connection toward the USB dongle used to communicate with the watch, by means of the SimpliciTI wireless protocol. The USB dongle receives and sends the SimpliciTI packets from/to the watch, while the driver processes and/or creates the packets to be used in the communication.

For example, when a button is pressed, the driver takes the packets received by the USB dongle, processes them in order to retrieve the relevant data (i.e., which button is pressed) and generates the corresponding notification event in Dog. The same process happens for the data coming from all watch sensors.

Upon receiving a command from Dog, the driver prepares the packets containing any message to write on the dWatch display and to turn on/off the buzzer integrated in the watch, in case of reminders. The USB dongle sends these packets to the watch, whose custom firmware acts accordingly, i.e., by converting the text into the seven-segments format and by showing the sent message on two lines, with scrolling text, etc., according to the specific active functionality and the length of the message.

The dWatch driver receives the accelerometer data from the watch and sends them to a bundle responsible of their correct interpretation for the gesture recognition process. The gesture recognizer is based on the computationally simple *uWave Gesture Recognizer* [13], proposed by Rice University and Motorola Labs. Once a gesture is correctly recognized, the dWatch driver sends the proper command to the device being controlled. The currently available set of recognized gestures (in Table 1) is still minimal and needs to be validated by a user study, jointly with the reliability of the gesture recognition algorithm used in this context.

6. Conclusions

In this paper, we introduced a personal wrist watch to support unobtrusive communication with a smart environment. The watch makes use of an off-the-shelf hardware that is easily obtainable and capable of rich interactions by using its sensors, integrated with different brands of home automation plants thanks to a multi-protocol open source home gateway. A wrist watch is very attractive for a domotic system, since the current tools for interacting with the environment are multi-purpose devices, that can be busy in other activities, that are not always carried around by users and that needs to be picked up and turned on before they are ready to be used.

The presented scenarios show different situations where the users may benefit from a wearable platform for home automation systems, by overcoming the depicted limitations of contemporary domotic solutions and trying to lower the technology gap that may be introduced by such systems.

Future work will extend this platform and validate it with extensive user studies, to better shape and prioritize the functionalities of such a watch, to evaluate the effectiveness of the proposed gestures and the performance of the adopted gesture recognizer.

Acknowledgements

This work was partially supported by Regione Piemonte under project STORIES of Polo di Innovazione ICT. Luigi De Russis currently exploits a research grant by the Lagrange Project of the CRT Foundation with the scientific coordination of ISI Foundation.

References

[1] M. T. Raghunath, C. Narayanaswami, User Interfaces for Applications on a Wrist Watch, Personal and Ubiquitous Computing 6 (2002) 17–30.

[2] J. Rekimoto, GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices, in: Proceedings of the 5th IEEE International Symposium on Wearable Computers, ISWC '01, IEEE Computer Society, Washington, DC, USA, 2001, pp. 21–27.

[3] D. Roggen, B. Arnrich, G. Trster, Life Style Management Using Wearable Computer, in: 4th International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications, 2006.

[4] U. Maurer, A. Rowe, A. Smailagic, D. Siewiorek, Location and Activity Recognition Using eWatch: A Wearable Sensor Platform, in: Y. Cai, J. Abascal (Eds.), Ambient Intelligence in Everyday Life, Vol. 3864 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, pp. 86–102.

[5] C. Harrison, B. Y. Lim, A. Shick, S. E. Hudson, Where to locate wearable displays?: reaction time performance of visual alerts from tip to toe, in: Proceedings of the 27th international conference on Human factors in computing systems, CHI '09, ACM, New York, NY, USA, 2009, pp. 941–944.

[6] T. Starner, The challenges of wearable computing: Part 1, Micro, IEEE 21 (4) (2001) 44 –52.

[7] A. Youssef, J. Krumm, E. Miller, G. Cermak, H. Eric, Computing Location from Ambient FM Radio Signals, in: IEEE Wireless Communications and Networking Conference, 2005.

[8] J. Pasquero, S. J. Stobbe, N. Stonehouse, A Haptic Wristwatch for Eyes-Free interactions, in: ACM Conference on Human Factors in Computing Systems, 2011, pp. 3257–3266.

[9] D. Bonino, F. Corno, L. De Russis, A user-friendly interface for rules composition in intelligent environments, in: P. Novais, D. Preuveneers, J. Corchado (Eds.), Ambient Intelligence - Software and Applications, Vol. 92 of Advances in Intelligent and Soft Computing, Springer Berlin / Heidelberg, 2011, pp. 213–217.

[10] D. Bonino, E. Castellina, F. Corno, The DOG gateway: enabling ontology-based intelligent domotic environments, IEEE Transactions on Consumer Electronics 54 (4) (2008) 1656–1664.

[11] D. Bonino, F. Corno, Dogont - ontology modeling for intelligent domotic environments, in: A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, K. Thirunarayan (Eds.), The Semantic Web - ISWC 2008, Vol. 5318 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2008, pp. 790–803.

[12] The OSGi Alliance, OSGi Service Platform Service Compendium, Tech. rep., OSGi Alliance (2009).

[13] J. Liu, L. Zhong, J. Wickramasuriya, V. Vasudevan, uwave: Accelerometer-based personalized gesture recognition and its applications, Pervasive and Mobile Computing 5 (6) (2009) 657 – 675.