POLITECNICO DI TORINO

DOCTORAL SCHOOL

Ph.D. in Computer and Control Engineering – XXVI cycle

Dissertation

Interacting with Smart Environments

Users, Interfaces, and Devices

Luigi De Russis

Advisor
Fulvio Corno

Ph.D. Coordinator
Pietro Laface

February 2014

Abstract

A Smart Environment is an environment enriched with disappearing devices, acting together to form an “intelligent entity”. In such environments, the computing power pervades the space where the user lives, so it becomes particularly important to investigate the user’s perspective in interacting with her surrounding. Interaction, in fact, occurs when a human performs some kind of activity using any computing technology: in this case, the computing technology has an intelligence of its own and can potentially be everywhere. There is no well-defined interaction situation or context, and interaction can happen casually or accidentally.

The objective of this dissertation is to improve the interaction between such complex and different entities: the human and the Smart Environment. To reach this goal, this thesis presents four different and innovative approaches to address some of the identified key challenges. Such approaches, then, are validated with four corresponding software solutions, integrated with a Smart Environment, that I have developed and tested with end-users. Taken together, the proposed solutions enable a better interaction between diverse users and their intelligent environments, provide a solid set of requirements, and can serve as a baseline for further investigation on this emerging topic.

Acknowledgements

Several influential individuals deserve a very grateful acknowlegment for their role in supporting me during these three years.

I am particularly grateful to my advisor, Fulvio Corno: I thank him for his support and mentorship. From day one, he has given me the freedom to pursue research that I was passionate about. With his guidance and patience, I grew from a fresh graduate student into a proper researcher. I must also acknowledge the other members of the e-Lite research group and, in particular, Dario Bonino for his constant support and friendship. Thank you!

I am also indebted to Bartolomeo Montrucchio, who had a significant impact on my teaching skills, and to all the people I have met in these years, even if only for a few moments.

Finally, I am forever grateful to my parents, Giovanni e Giuseppina, and my brother Davide, who have always supported and encouraged me. Thank you for all the lessons learned, stories shared and values instilled.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.
>
> *Mark Weiser*

> I think users want to have the feeling they did the job - not some magical agent.
>
> *Ben Shneiderman*

1.1 Overview

A Smart Environment (SmE) is a "small world where all kinds of smart devices are continuously working to make inhabitants' lives more comfortable." [1] Smart Environments represent a multidisciplinary area which stems from various fields of computer science and electrical engineering: artificial intelligence (AI), networks and sensors, ubiquitous computing (ubicomp), and human-computer interaction (HCI). Building upon such areas, a Smart Environment aims at providing flexible and intelligent services to users acting in an environment [2] where the technology disappears from sight and becomes integrated into the daily life in a way that people can use it without consciously thinking about it [3].

Thanks to the advancements in sensors capabilities, mainly arising from building automation and home automation areas, a SmE can benefit from a variety of different inputs, that range from simple sensor values (e.g., on/off, temperature) to more complex data, like sound and images, also including context-aware information (such as user location or identity). Such inputs, coming from different and distributed devices, are combined by an “intelligent”[1] system, that employs AI-related techniques (Semantic Web and intelligent agents [4], typically), to understand the current status of the environment and to possibly provide some type of support to its inhabitants [5, 2].

A SmE is composed of several artifacts and components, but its basic elements are users, a middleware software (or gateway), applications, devices, and the environment itself, as depicted in Figure 1.1 and better detailed below. With their interactions, the desired functionalities can be achieved. According to the adopted technologies and their application scenarios, in the literature such environments are mostly referred as Smart Environments (SmE), Smart Spaces, or Ambient Intelligence (AmI).

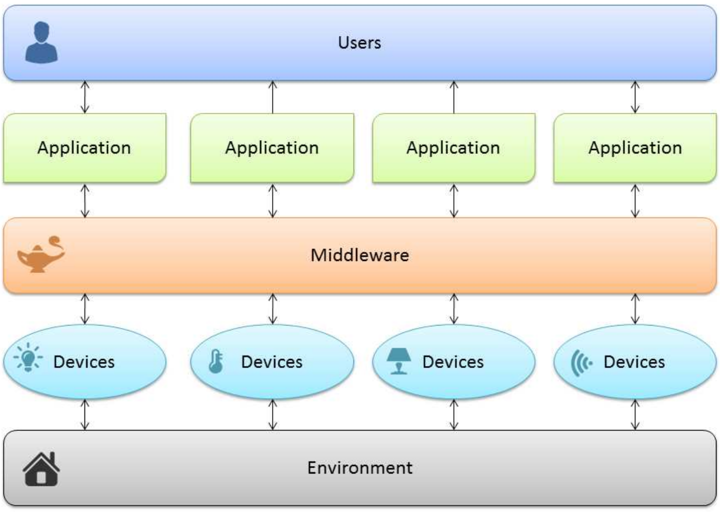


Figure 1.1. SmE Basic Elements and their Interactions

Envisioned **environments** extend from homes to offices, shopping malls, factories, and classrooms.

Devices can range from simple sensors (e.g., temperature sensors) to multi-feature devices (e.g., smart watches with multiple I/O capabilities), thus encompassing:

[1] the word “intelligent”, here, mostly refers to Artificial Intelligence, as defined by Norvig and Russel in [4]

- sensors for detecting or measuring motion, light, temperature, humidity, air quality, and other conditions that are descriptive of the environment;
- actuators for acting on doors, lighting systems, air conditioners, and other controllable parts of the environment;
- wearable devices that are positioned on the body, needs to run continuously, to be operated hands-free, and can monitor features that are descriptive of a person's physiological state or movements; and
- other end-user devices, such as computers and mobile devices, for providing particular input and output capabilities and network connection.

Applications are strongly dependent on the type of the environment, the number and type of devices included in it, and the users needs and capabilities. Each application typically represents one or more specific service domain in an established environment for a given category of users.

The **middleware** includes the "intelligent" part of the system that is responsible of: (a) merging the flow of data coming from the devices through wired and/or wireless networks; (b) making them more useful to other applications present in the system, by performing some processing operations; and (c) using advanced techniques, such as AI, for taking decisions or supporting the environment inhabitants. Moreover, an important task typically assigned to the middleware is to facilitate interoperability, i.e., to help devices and networks created by different providers and speaking different protocols to effectively understand each other and converge into a unique and uniform representation that can be easily understood by other actors in the SmE [2].

Finally, **users** are at the center of a SmE [6, 7, 8, 9]: a smart environment should be able to help people of all ages, conditions and educational background with a proactive but sensible attitude, i.e., by keeping a balance between not missing an opportunity when the user expected assistance, and meanwhile to refrain from interfering with its inhabitants when it is not required ot would not be appreciated. On one hand, in fact, there is a will to reduce explicit human-computer interaction as the system is supposed to use its "intelligence" to infer the situations and user needs from the collected data, to help when (and only if) required. On the other hand, a diversity of users may need or voluntarily seek direct interaction with the system to indicate preferences, needs, etc.

A fundamental principle of SmE is, in fact, that users should be always in control [10, 11] and should be able to decline any advice coming from the system, impose their preferences, undo any decision and action taken autonomously from the system and even shut down the system if it is perceived as inconvenient [5]. For this reason, it is mandatory to find a proper balance between the system that autonomously acts and informs, and the system that is directly controlled and managed.

1.2 Motivation

What is clear from the definition and principles of SmE is that people and their activities are central to the realization of its vision. Whatever constitutes the "intelligence" of a SmE, it has to reveal itself to users through human senses and must be geared towards human comprehension.

The vision of SmE changes the established relation between humans and technology: the user is interacting with an environment that comprises various disappearing devices, acting together to form a smart "entity", and no longer with a computer or a mobile device.

According to Butz [2], the interaction bandwidth between a person and a SmE is "much higher" than traditional interactions with computers, since the "*human user can interact using her* [. . .] *entire body and multiple senses*". In a SmE, in fact, multiple devices with different input and output capabilities, and various users with diverse needs coexist. Each user can interact at any time with the environment in an unpredictable manner, more like the way people interact among themselves and with the physical world. Users should be able to interact with a SmE in such a multimodal way, according to their physical and mental capabilities: classical HCI paradigms, such as WIMP (Windows, Icons, Mouse, and Pointer), are not enough for responding to all the possibilities that can exist. People speak, gesture, see, and use writing tools to communicate and to alter physical artifacts.

Supporting human activities, thus, imply a variety of important changes to input, output, and interactions that define our experience with computing. The communication from the user to the environment (i.e., input) tends to become "implicit", moving beyond textual input from keyboards and selection from mice to a higher variety of data types and input technologies, encompassing our natural interaction with the physical environment. The communication from the environment to the user (i.e., output) becomes "distributed" and available in many modalities (e.g., auditory, speech, visual), locations (e.g., monitors, TVs, projected on walls) and forms (e.g., on wearable devices or through the actuation of some ambient lights). All interaction, in general, should be as unobtrusive and "natural" as possible: users have to interact with the smart environment in different, explicit and implicit, alternative or combined ways, to ensure a robust and natural interaction. Moreover, users should not perceive the system as an opponent, an "entity" whose goal is to reduce their autonomy and control over the environment.

Up to now, the main focus of SmE research area has been on technology development (i.e., sensors technology, networking, middleware, users localization and identification, etc.) and on solving security and privacy concerns that may enable a real usage and deploy of such environments. As a consequence, unfortunately, insufficient HCI practices, theories and models exists in SmE [9], and an effective interaction between users and the smart environment is still a challenging aspect [7, 5].

We may justify the last statement with a literature research. We select the IEEE Xplore Digital Library and we query it[2], looking for papers related to *smart environments* and *smart homes*. This search finds 2101 results, counted after cleaning the results from unrelated papers and excluding keynote talks, tables of contents, front covers and similar content. An informal topic analysis of this preliminary investigation seems to confirm the technology-related focus of SmE research area: the main topics emerging from such results, in fact, encompass activity recognition, autonomous interaction between intelligent services and devices, user localization, and algorithmic improvements of intelligent systems (by using multi-agent or Semantic Web based techniques, typically).

If we narrow the search to papers that explore and deal with some forms of interaction between the user and the environment, we obtain 188 results[3], only, less than 9% of the initial set. Most of these papers present user interfaces of various types, from natural interaction to mobile-based interfaces, and in diverse contexts. However, they are not built upon a proper SmE system, as in a typical SmE architecture (depicted in Figure 1.1), but they are deeply integrated inside a specific middleware or intelligent system, becoming one of multiple functionalities offered by the middleware itself. Moreover, typically, they do not offer a base set of requirements to reproduce the application with other systems and thus validating the user-related results.

This thesis aims at improving the interaction between users and Smart Environments, by exploring challenging and different approaches in key areas and providing a set of tools and applications, loosely coupled with the underlying middleware, that can serve as a baseline for further investigation.

Such approaches and applications are based on solid and explicit requirements, extrapolated from the literature or obtained by investigating users.

1.3 Contribution

User-related research in the SmE area mainly focuses on activity recognition, considering users localization and movements inside the environment and taking into account their privacy and security. Emerging trends encompass distributed user interfaces, systems to promote users' behavior changes, natural interaction, and applications for keeping users as autonomous as they desired for controlling a SmE.

[2] search made on February 22, 2014, with the following query: `"smart environment" OR "smart environments" OR "smart home" OR "smart homes" OR "ambient intelligence"`

[3] using the query: `("smart environment" OR "smart environments" OR "smart home" OR "smart homes" OR "ambient intelligence") AND ("interaction" OR "interacting" OR "user interface" OR "hci")`, and cleaning the results as before

This thesis will explore some of these trends, with the main and final goal to *improve SmE interaction* (see Figure 1.2 for an overview). To reach such a goal, four approaches are being tackled, in different domain areas, targeting the home as the envisioned environment. They follow two different but parallel paths: improving smart environment interaction by lowering access barriers, and by keeping the human in the loop.

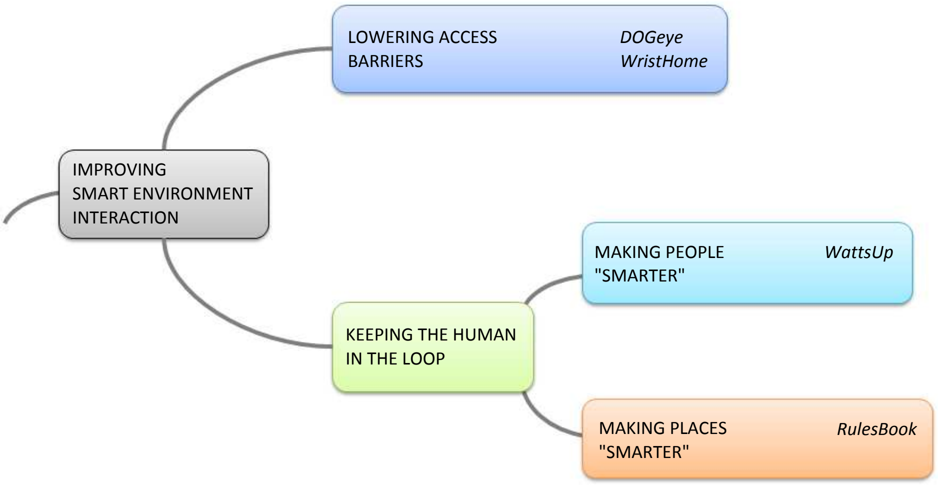


Figure 1.2. Contributions of this Thesis

Each approach is associated to a specific software application that exemplify and realize it. The applications explore four different paradigms in their paths.

Universal Access for lowering access barriers is introduced by DOGeye, a multimodal eye-based application for home management and control, based on state-of-the-art technologies in both eye tracking and home control, and aimed at people with severe and evolving impairments, such as ALS (Amyothrophic Lateral Sclerosis).

Wearable Computing for lowering access barriers is presented by WristHome, a system for turning existing wrist-worn devices into flexible home access points by exploiting a modular architecture independent from the specific watch device and from the home automation system.

Activity Delegation for making places “smarter” and keeping the user in control is presented by RulesBook, a mobile application for the creation of rules and context-aware applications by end users. The objective here is to let the user maintain the desired autonomy in her home; a common way to realize this goal is by explicitly delegating some tasks to the smart environment, thus perceiving it as useful and cooperative.

Behavior Change for Energy Saving for making people more “aware” is exhibited by WattsUp, a system for informing users about their energy consumption patterns, and by suggesting more efficient and “green” behaviors. Such a system is supported by a user survey, distributed online between September 2010 and January 2011, and completed by 992 people.

The main focus in this dissertation is on users and their interactions with the overall system, and not on the “intelligent” part. This lets the applications be general enough to be easily replicated with various middleware or “intelligent” systems, without invalidating the obtained results. The current implementation of such applications use an existing middleware software named Dog [12], as the “intelligent” part of the system.

1.4 Organization

The remainder of this thesis is organized as follows:

Chapter 2 summarizes related applications stemming from disparate but related domains, including Universal Access, Wearable Computing, Activity Delegation, End-User Programming, and Energy Feedback.

Chapter 3 presents the “founding technologies” that constitute the typical smart environment where the applications discussed in the following chapters are applied. Such technologies are Dog, a middleware actively developed and maintained by the e-Lite research group of the Politecnico di Torino, and DogOnt, an ontology that may provide a reasoning ground for supporting “intelligent” behaviors to the gateway.

Chapter 4 presents DOGeye, a multimodal eye-based application for home management and control, based on state-of-the-art technologies in both eye tracking and Smart Environment, and aimed at people with severe and evolving impairments, such as ALS (Amyothrophic Lateral Sclerosis).

Chapter 5 introduces WristHome, a system for turning existing wrist-worn devices into flexible home access points by exploiting a modular architecture independent from the specific watch device and from the particular home automation system.

Chapter 6 is about RulesBook, a mobile application for the creation of rules and context-aware applications by end users, to delegate some tasks to the smart environment, so that they can easily choose their preferred and personalized level of autonomy.

Chapter 7 describes the findings from a user survey about home energy consumption feedbacks, and the behavior changes they could infuse in the home inhabitants. Moreover, it provides a first validation of a user interface design for realizing the envisioned survey objective.

Chapter 8 follows the work introduced in the previous chapter and presents WattsUp, a system for informing users about their electrical power consumption patterns, and by suggesting more efficient and "green" behaviors.

Chapter 9 concludes the thesis and provides an overview on possible future works.

Chapter 2

Related Systems

My research on improving interaction with SmE draws from a variety of areas, including interaction techniques, visualization, wearables, accessibility, and end-user programming. Here I focus on related works that was most influential and relevant, introducing them according to the main contribution paradigms presented in Chapter 1.

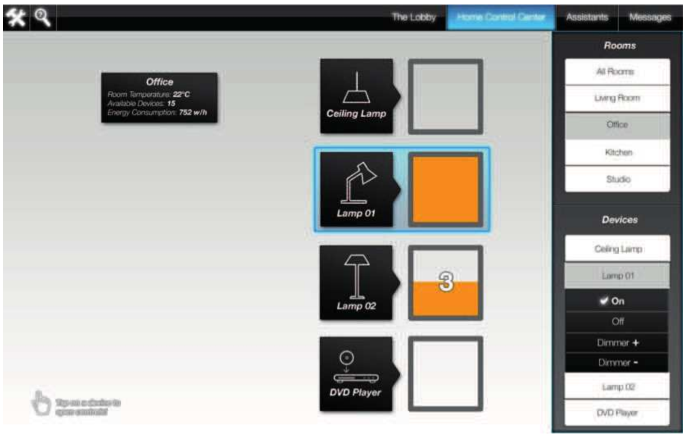
2.1 Universal Access

While SmE technology is maturing and evolving, a sensible lack of user interfaces for controlling such environments using eye movements can be easily spotted. Gaze-based applications are particularly interesting for lowering the access barrier to SmE since they are well-suited for people with severe and progressive impairments, such as ALS. Given the particular capabilities of the target users, gaze-based applications for SmE should follow well-defined requirements, such as the *Gaze Based Environmental Control* guidelines proposed by the COGAIN European project (reported in Chapter 4 and defined by Corno et al. [13]). Such guidelines are divided in four categories:

- *Control applications safety*: guidelines concerning the behavior of the application in critical conditions, such as alarms and emergencies.
- *Input methods for control application*: guidelines about input methods that the control applications should support.
- *Control applications significant features*: guidelines impacting the management of commands and events within the house.
- *Control applications usability*: guidelines concerning the graphical user interface and the interaction patterns of the control applications.

Overall, different studies and surveys about HCI perspective on Smart Homes are present in literature, such as the work of Hee-Cheol Kim et al. [14]. In the same way, a variety of user interfaces are proposed for controlling a smart environment using different traditional input modalities, such as mouse, keyboard and remote control [15, 16].

Natural interactions in a SmE are possible thanks to application like the *Home Operating System* (Home OS) [17] (Figure 2.1), a multimodal interface proposed by the Technical University of Berlin. HomeOS allows users to control their homes using touch, speech and gesture interactions. However, according to the cited COGAIN guidelines, this application presents some issues, and eye tracking interaction was not considered at all.



Office
Room Temperature: 22°C
Available Devices: 15
Energy Consumption: 752 w/h

Figure 2.1. The Home Operating System

Commercial eye tracking systems, from the other side, typically include graphical interfaces for environmental control, obviously based on gaze interaction. *LC Technologies* provides a basic eye-controlled “Light and Appliances” interface (Figure 2.2), bundled with some electrical switching equipments. This system provides basic control of light and appliances located anywhere in the home. The user can turn appliances on and off by looking at a bank of switches displayed on the screen, with commands sent to home sockets and lights via the home mains electricity wiring, exploiting the X10 home automation protocol.

The same basic functionalities are offered by the DynaVox LifeMate module named *EnviroMate* (Figure 2.3(a)) and by the SR Labs iAble module named *DOMOTICS* (Figure 2.3(b)). EnviroMate presents a grid of buttons related to environmental objects and can use X10, Z-Wave and general purpose IR for interface-to-device communications. The DOMOTICS module of the iAble software takes a pretty similar approach: it displays a grid of buttons and communicates with home devices through an infrared transmitter (supplied with the system) or via X10. Both

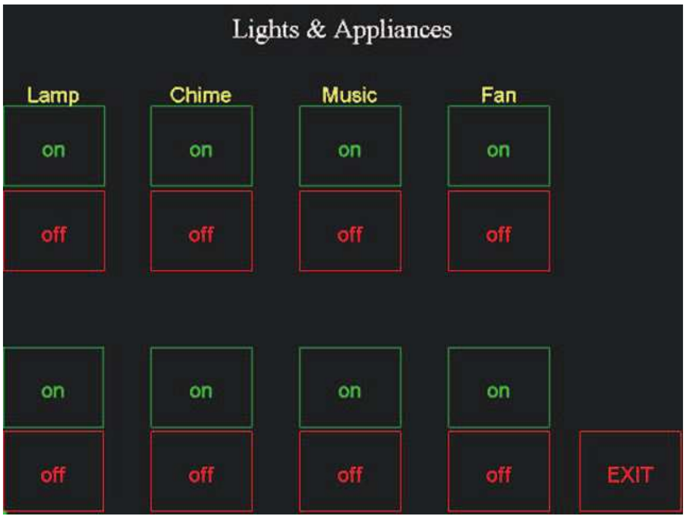
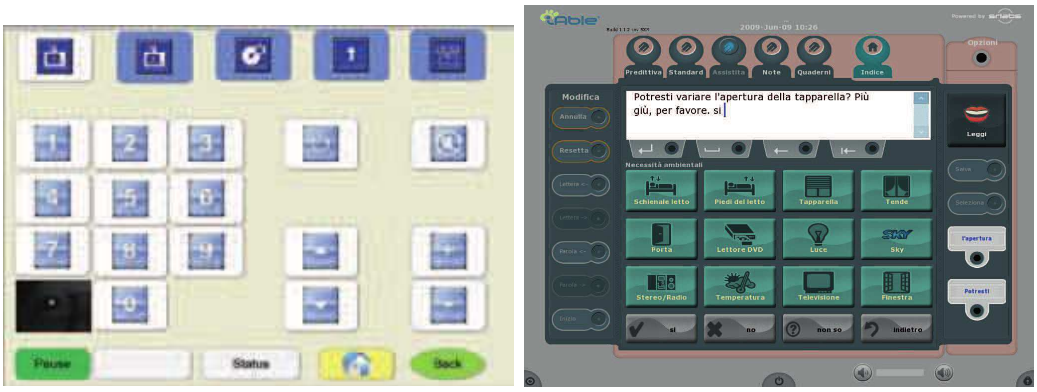


Figure 2.2. LC Technologies domotic interface

interfaces are quite simple and incomplete in their functionalities, according to the COGAIN guidelines.



Potresti variare l'apertura della tapparella? Più giù, per favore. si

(a) LifeMate EnviroMate

(b) iAble DOMOTICS

Figure 2.3. EnviroMate and iAble applications

HomeOS, LifeMate EnviroMate and iAble DOMOTICS has been evaluated against the COGAIN guidelines by both analyzing nominal features and runtime execution of the three applications. Every guideline category has been considered separately and, for each category, the number of satisfied guidelines versus the total number of category guidelines has been measured. If an application does not respect any guideline in a category, its COGAIN support level is labeled as “Absent” for the category; if part of the guidelines (but not all) are respected, for a given category, the application is labeled as having “Partial” support for the COGAIN guidelines in the considered category. Finally, applications respecting all guidelines in a given category are labeled as providing "Full" compliance.

Table 2.1 summarizes the evaluation results showing that none of the considered applications have full support to the COGAIN guidelines and, moreover, that partial support is actually very limited, being less than 50% of the guidelines on average.

Guidelines category	**Home OS**	**EnviroMate**	**iAble DOMOTICS**
1. Safety	Absent (0/5)	Absent (0/5)	Absent (0/5)
2. Input methods	Partial (3/5)	Absent (0/5)	Absent (0/5)
3. Operative features	Partial (3/6)	Partial (2/6)	Partial (2/6)
4. Usability	Partial (2/5)	Absent (0/5)	Absent (0/5)

Table 2.1. An evaluation of the applications presented in this Section. The numbers in parenthesis represent the quantity of fulfilled guidelines per category.

2.2 Wearable Computing

Wearable computing can play an important role in the design of SmE, especially when the two fields can be easily integrated. As stated by Cook and Song [18], research in wearable computing and research in smart environments has been pursued independently, even if these disciplines can benefit each other. For example, fusing data from worn sensors and from passive environmental sensors can facilitate the creation of more comprehensive and more accurate models of resident behavior and well being. In addition, information collected in the environment can be used to predict resident physiological response (validated by wearable sensors) and information collected from wearable sensors can be used to initiate appropriate responses and changes in the environment.

Wearable devices can also facilitate the interaction with smart environments, lowering the access barrier to such environments. Typical interaction with a SmE, in fact, happens by means of fixed, in-home, touch panel or with software applications for computer or mobile devices. However, there are situations where is not possible, secure or suitable to use one of these devices, for example with wet hands, or in situations where hands need to be free. In particular, an interesting wearable object is a wrist watch (o bracelet). It has a form factor that makes it highly available and unobtrusive to the user and it can be instantly viewed with the flick of the wrist. Moreover, a wrist watch is very attractive for four main reasons [19]:

- a large fraction of the population is already accustomed to wearing this type of devices;
- watches are less likely to be misplaced compared to mobile devices;
- watches are more accessible than other devices one may carry;
- a wrist watch is ideally located for body sensors and as a wearable display.

An interesting wrist watch that can be used in a smart environment is the *WristQue*, designed and developed by the MIT Media Lab [20]. WristQue combines environmental and inertial sensing with precise indoor localization data for providing the user with a personal control interface. Devices can be controlled with gestures and selected by pointing them. Even if it is a promising device, current applications consist only in controlling light intensity by using one of the WristQue buttons or some gestures. Future works will encompass data fusion with infrastructure-based sensors and additional control inputs (such as tapping or scratching the watch).

The eWatch [21], jointly developed by Carnegie Mellon University (USA) and Technische Universität München (Germany), 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.

Between smart watches that are not thought for interactions with SmE, the IBM Linux Watch [22] can be considered the first one: 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 interact with any smart environment.

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). 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 been run until January 1st, 2012, when it has been totally discontinued.

The eZ430-Chronos[1] watch is an off-the-shelf, programmable wrist watch produced by Texas Instruments. It may be used as a reference platform for watch

[1] `http://processors.wiki.ti.com/index.php/EZ430-Chronos`, last visited on January 2014

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 or its integration with a commercial gateway for home automation, based on the Z-Wave protocol.

Finally, the Pebble[2] smartwatch is a companion for Android and iOS smartphones. It features an e-paper display with LED backlight, it is water-proof, a long lasting battery life and provides a 3-axis accelerometers, an ambient light sensor and a Bluetooth 4.0 connection. It comes with a SDK for realizing custom applications to be uploaded on the watch itself, thus enhancing the smartwatch capabilities. Currently, it is not employed in any SmE system, but soon it will interface with the home automation system proposed by iControl.

2.3 Activity Delegation

According to the vision of SmE, a fundamental principle is that users should be always in control of the environment. To realize this principle, a proper balance between the actions carried on by the "intelligent system" and the direct user control has to be find. A relatively accepted approach to find this equilibrium is *Activity Delegation*, i.e., users should be able to delegate some activities to the "intelligent" system, thus obtaining a personalized and easily changeable level of autonomy. Activity Delegation is usually performed by defining some *rules* (also known as "Contex-Aware Application", such as in [23]) that have to be set up by end-users, i.e., people without a specific knowledge or technical skills.

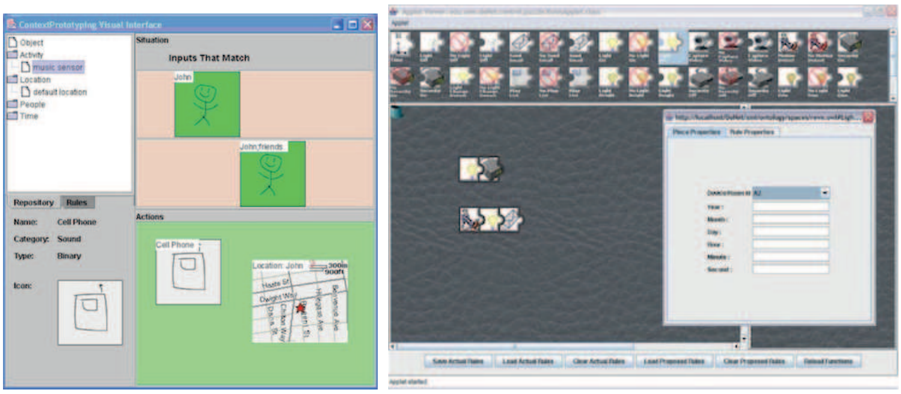
Different rule composition interfaces, aimed at end-users, are present in the literature, either for Smart Homes or for other application domains. An example of a linear and simple interface, oriented to game programming, is the *Rule Editor Pane* included in the GameSalad application, a game creation tool explicitly designed for non-programmers. Such an editor permits to create rules by expressing alternative behaviors that can be applied to game actors under certain conditions. For example, it allows to compose rules such as "When mouse button is down, display text: *Try to take over the world*; otherwise, display text: *What are we doing tomorrow?*".

OpenBlocks [24] was a general-purpose framework in Java for graphical block

[2] `http://getpebble.com`, last visited on January 2014

programming systems, developed at the MIT. This framework presents a great expressiveness and could be used to realize a rule builder. However, OpenBlocks requires a heavy customization to be adapted for a SmE, and its higher expressiveness can lead to increased complexity for rules creation. Similar to OpenBlocks, *blockly* is a web-based, graphical programming editor developed and maintained by Google.

iCAP [25], the rule editor of DoNet [26] and GALLAG Strip [23] are, instead, three examples of applications specifically developed for defining rules in SmE.



Name: Cell Phone

Category: Sound

Type: Binary

Icon:

(a) iCAP interface

(b) DoNet Rule Editor

Figure 2.4. iCAP and DoNet

The *iCAP* interface has one window with two main areas (see Figure 2.4(a)). A tabbed window (on the left) is the repository for user-defined inputs, outputs, and rules. The input and output components are associated with graphical icons that can be dragged into the right area, then be used to construct a conditional rule statement. The right area contains the two elements of a conditional rule statement (antecedent and consequent). An example rule could be: *IF Sam is in the office after 5pm and the temperature is less than 10 Celsius degrees OR IF Jane is in the bedroom and the temperature is between 0 and 15 Celsius degrees, THEN turn on the heater in the house.* The upper side of the right area represents the "if" portion of the rule, and can be split into one or more "sheets". Inputs on a single sheet are related by conjunction and multiple sheets are related by disjunction. The bottom side of this area represents the "then" portion of the rule condition. The main drawbacks of iCAP are the requirement for a user to draw each object she wants to use in a rule, its pen-based nature, and the narrow separation between the concepts of event and constraint.

The rule editor of *DoNet* is based on the context-aware *Jigsaw Editor* [27] application (Figure 2.4(b)). Each puzzle piece corresponds to an object to be used in a rule and has a set of properties, editable through a properties panel. This editor interface is sub-divided into three main areas consisting of puzzle piece choice panel, the proposed rules panel and the actual rules panel, to provide a drag and drop rule building environment. The main drawbacks of such an editor are the limited expressivity of its language and the need to open a dedicated panel to set or view the details of each rule part.

Eventually, *GALLAG Strip* (Figure 2.5) is a tool aimed at the creation of context-aware applications (i.e., rules) for helping people to break bad habits and challenge their behaviors in SmE.

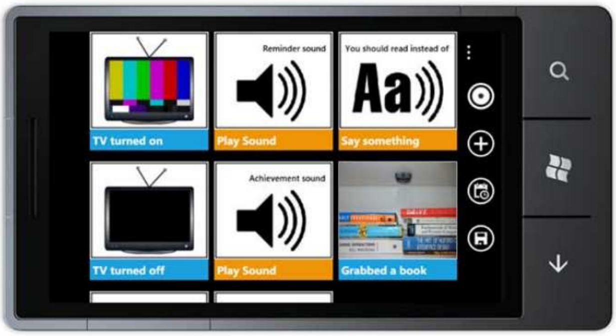


Figure 2.5. GALLAG Strip

GALLAG Strip has the peculiarity to use a "tangible" approach to end-user programming, i.e., it enables programming by physical demonstration of envisioned interactions with the same sensors and objects that users will later encounter in their completed rule.

2.4 Behavior Change for Energy Saving

Home energy consumption and related user behaviors are currently being studied by several research groups worldwide, with the aim of understanding how home inhabitants consume energy and with the goal of finding new interactions and habits in the home able to encourage more energy-efficient behaviors. In this context, research studies mainly involve: house occupant characterization [28], and behavior modeling [29, 30, 31, 32], mining and simulation of typical consumption profiles [33, 34], rule-based management systems for reducing consumptions of daily activities [35] and feedback interfaces and monitors able to "persuade" users to modify or adapt their habits to achieve increased savings [36, 37]. These various effects (user habits, automation,. . .) can be fruitfully combined in real settings, but we believe they are best analyzed separately.

As proven by many research pilots and surveys [28, 38, 39, 40], energy feedback is primarily a human-related task needing user centered approaches for being tackled. Different kinds of feedbacks may be employed and they can either induce changes into home inhabitants habits or be completely ignored depending on many factors including users' green attitude, visual appearance, understandability of exposed data, etc. Among investigated mechanisms and visual solutions, the research community has currently reached a partial consensus on a set of basic interactions that are generally successful in promoting reductions in energy consumption. These solutions include:

- goal setting interfaces, i.e., interfaces based on users' desire of fulfilling a given (energetic) objective, either induced by the interface or self-imposed by home inhabitants;
- direct feedback, i.e., timely updated in-home displays (IHDs) showing the home current energy consumption;
- historical trends in consumption, showing how home consumption evolves over time and highlighting temporal correlations, e.g., in northern countries the winter season usually has higher consumptions;
- non-obtrusive displays, i.e., displays designed to weave themselves into the home environment, attracting the user attention when needed but avoiding intrusive settings and interactions that may foster interface abandoning or disposal.

Unfortunately, these interaction paradigms have been widely but sparsely investigated, and few approaches can be found, in the literature, that focus on the complete design process of IHDs, by applying user centered design principles from the early stage (interaction) to the final in-home deployment [36].

The 2004 survey on "Consumer preferences for improving energy consumption feedbacks" [41] is one of the earliest works in this field. In this survey, carried by Simon Roberts, Helen Humphries and Verity Hyldon, focus group research is used to assess consumer preferences for feedback and improved information about energy consumption at home. A series of 7 focus groups in three different parts of England were held, dividing groups by bill payment methods. The study findings showed typical behaviors of home energy consumers, reporting interesting insights on the energy behaviors of the interviewed householders. In particular the study showed that home inhabitants:

- exhibit a high level of cynicism about the motivations of energy suppliers to promote energy saving and a general distrust in their advice;

- show high awareness and knowledge of energy saving measures and techniques but do not know the cost, and assume it is very expensive;
- demonstrate little motivation to act and high resistance to being forced to act;
- have very clear preferences (and dislikes) on feedback options;
- would, given the right feedback, examine reasons for change in consumption and may be stimulated to take actions.

Out of the focus group responses, equally strong preferences emerge for simple bar charts with historical data and direct consumption visualization. With respect to the survey reported in this paper, results are somewhat overlapping, showing users preferring simple and clear feedbacks. However the two works cannot directly be compared since the Roberts survey was mainly focused on paper-based feedbacks while in this study we are more concerned on real-time energy feedbacks.

G. Wood and M. Newborough [42] investigated the energy use information transfer in the home, with the aim of better enabling/fostering energy conservation through central and local displays. In their work, they analyze and discuss methods for motivating energy-saving behaviors and for presenting energy-use information on two different kinds of in-home displays. According to Wood studies, information alone about energy use in a room, by an appliance, in a time period, by an end user or during an activity will not motivate energy saving. Experimental evidence showed, in fact, that such information needs both to be displayed in a simple manner and appropriately grouped in order to motivate home inhabitants. Among several feedback opportunities, Wood and Newborough reported goal setting, self competition and monetary rewarding as the most effective interactions. On the converse, they demonstrated that expressing energy use in monetary units is not effective due to the small daily cost of consumed energy. Similar ineffectiveness is also shown by carbon dioxide and other environmental units to which home inhabitants are not accustomed, while the classical kWh energy measure is better accepted, although few people really understand this unit.

Nevertheless, the effectiveness of in-home energy displays (IHD) is confirmed by the survey carried by Faruqui et al. [43] in May 2009. Faruqui et al., economists working with the Brattle Group, reviewed a dozen of pilot programs in North America, and abroad, either focusing on energy conservation impact of IHDs or that studied demand-side management tools and include IHDs as one of the tools. They also reviewed customer opinions and attitudes towards IHDs and direct feedback. Results show that direct power feedback provided by an IHD actually encourages people to make more efficient use of energy. Moreover, in their study, Faruqui et al., found that IHDs can reduce consumption of energy, on average, by about 7% when pre-payment of energy is not involved. Instead, when users are using IHDs and electricity prepaying systems they can reduce energy consumption by roughly 14%, on average. This confirms the increasing research attention on direct and real-time energy feedback systems.

On the same topic, Sarah Darby [44] carried a literature review on metering, billing and direct displays with the aim of better understanding the effectiveness of energy feedbacks to householders. According to Darby, overall literature demonstrates that clear feedback is a necessary element in learning how to control energy consumption more effectively over a long period of time, and that instantaneous direct power feedback in combination with frequent, accurate billing is needed as a basis for sustained demand reduction. Savings resulting from energy consumption feedback range between 5% and 15% in case of direct power feedback (the focus of the survey presented in this paper) and between 0% and 10% for indirect feedback, i.e., billing. According to Darby, user-friendly displays are needed as part of any new meter specification. Monitors will be most useful if they show instantaneous usage, expenditure and history feedback as a minimum, with a potential for showing information on micro-generation, tariffs and carbon emissions.

Besides energy efficiency, designing and evaluating IHDs has a strong human component which is currently attracting several efforts from the human-computer interaction research field. In the last years always-on electricity feedback, and implied issues, gained momentum in this community leading to several interesting approaches. Riche, Dodge and Metoyer [45], for example, conducted a study to understand consumer awareness of energy consumption in the home and to determine the requirements for interactive, always-on IHDs to gain awareness of home energy consumption. They then designed a three stage approach to support electricity conservation routines based on raising awareness, informing on complex changes and maintaining sustainable routines. Although not statistically significant, since the user group was too small, the results of their study highlighted several design suggestions/implications including the potential of location-based feedback for providing awareness and the necessary compromise between readability and aesthetics in always-on feedback.

Tae-Jung Yun investigated the impact of a minimalist IHD [37] showing that even very simple visual feedback may have an impact on household consumption when combined with self-goal setting strategies on the part of the user, without any explicit goal setting interface. However, minimal solutions do not meet the needs of users who consider themselves to have high awareness of energy consumption in their homes, requiring more sophisticated interfaces.

Psychological implications of energy displays and interaction paradigms may also influence the effectiveness of IHDs as demonstrated by the studies of He and Greenberg [38] and of Pierce et al. [39] remarking the importance of gathering, analyzing and responding to actual user needs during the design of feedback solutions.

Chapter 3

Founding Technologies

This chapter presents the "founding technologies" that constitute a typical smart environment where the approaches discussed in this thesis are applied. Such technologies are Dog, a middleware software actively developed and maintained by the e-Lite research group of the Politecnico di Torino, and DogOnt, an ontology that provides "intelligent" behaviors to the gateway.

For the purpose of this thesis, the presented applications can be considered loosely coupled from such "founding technologies": Dog and DogOnt, in fact, are only used for providing access to the smart environment capabilities.

3.1 DogOnt

DogOnt is a domain ontology (i.e., "an explicit specification of a conceptualization", as defined by Gruber [46]) specifically designed to model smart homes equipped with commercial domotic plants and intelligent appliances (for a complete description of the DogOnt design and modeling capabilities see [47]). It is encoded in the OWL format, as suggested by the W3C Semantic Web standard. It is currently exploited by the open source Dog gateway [12], and it is organized along 5 main hierarchies of concepts (Figure 3.1, hierarchy roots in bold) supporting the description of:

- the domotic environment structure (rooms, walls, doors, etc.), by means of concepts descending from *BuildingEnvironment*;

- the type of domotic devices and of smart appliances (concepts descending from the *Controllable* subclass of the *BuildingThing* main concept);

- the working configurations that devices can assume, modeled by *State*s and *StateValue*s (see the following paragraphs for more details);

- the device capabilities (*Functionalities*) in terms of accepted events and generated messages, i.e., *Command*s and *Notification*s;
- the technology-specific information needed for interfacing real-world devices (*NetworkComponent*) and
- the kind of furniture placed in the home (concepts descending from the *UnControllable* subclass of the *BuildingThing* main concept).

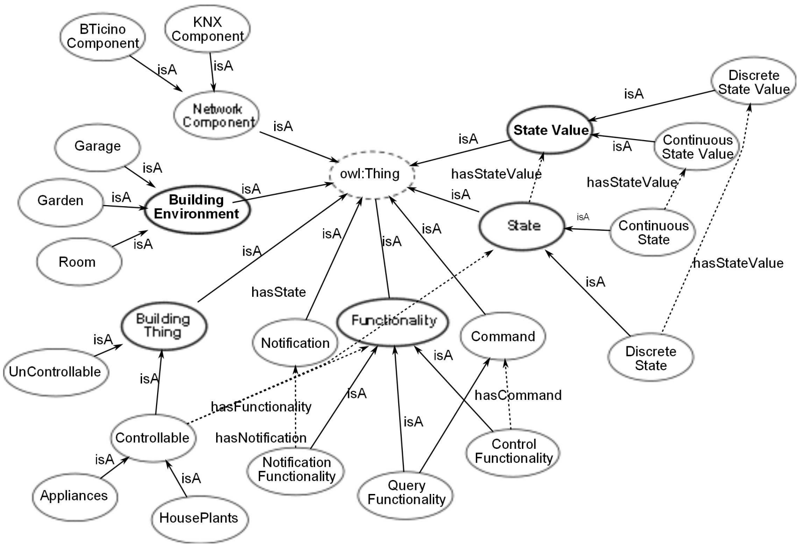


Figure 3.1. DogOnt in a nutshell

DogOnt models home automation devices in terms of functionalities and states.

Functionalities They describe the device under the viewpoint of device interaction capabilities, i.e., they describe how a given device can be controlled, queried and whether it can autonomously generate “events.” For example, while a lamp can only be switched on and off, a light sensor can either be queried for the current luminance or can autonomously send luminance change events at regular time intervals. DogOnt functionalities include:

- *ControlFunctionalities*, modeling the ability of a device to be controlled by means of some message or command,
- *QueryFunctionalities*, modeling the ability of a device to be queried about its current state, and
- *NotificationFunctionalities*, modeling the ability of a device to issue notifications about state changes, in an event-driven interaction model.

Functionalities are either associated with commands (for *ControlFunctionalities*) or with notifications (*NotificationFunctionalities*) that further detail the specific operations supported by DogOnt device instances. Figure 3.2 shows a sample DogOnt model of a dimmer lamp, with functionalities highlighted in bold.

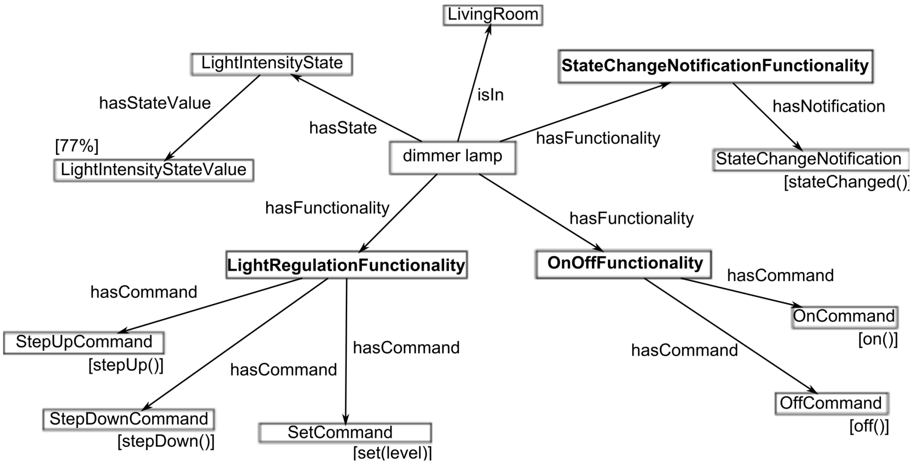


Figure 3.2. A sample Dimmer Lamp model in DogOnt

Device interconnections are modeled by the *controlledObject* relationship linking a controller[1] device (e.g., a switch) to one or more controlled devices[2](e.g., a group of lamps). The same device can be involved in different connections with different roles, i.e., as either a controller or a controlled device. Connections can be further specialized through the *generatesCommand* relation, which permits to specify the command(s) generated in response to a given device notification (Figure 3.3).

States They describe the various stable configurations that a device can assume during its working life-cycle. From the modeling point of view, each device may include one or more different simultaneous behaviors. If we refer to a CD Player, it can either be on or off, it can be playing a CD track with a given number and, it may have a specific earphone output volume. In DogOnt such behaviors are called `dogont:States`. The description of each `dogont:State` is represented by a set of identifiers, called `dogont:StateValue`, that model each operating condition. For example the CD player is modeled as having three independent `dogont:States`: a `dogont:OnOffState`, a `dogont:PlayingState` and a `dogont:VolumeLevelState`. Each of these three states include a specific set of possible state values (for

[1] `rdfs:domain(dogont:Control)`

[2] `rdfs:domain(dogont:Controllable)`

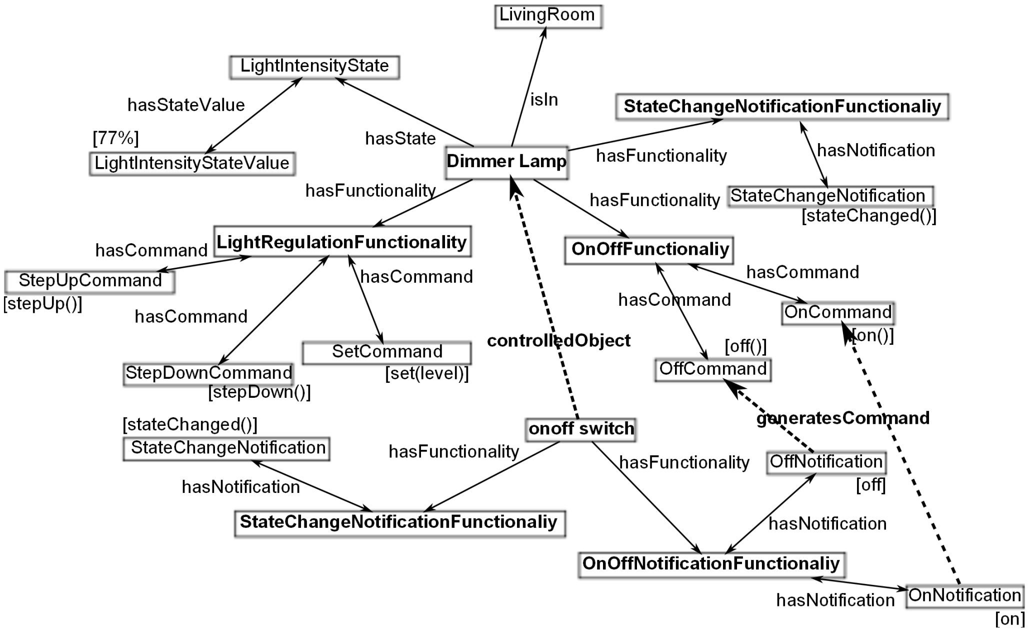


Figure 3.3. A sample of connection modeling in DogOnt where a Switch controls a Dimmer Lamp

example, the first state includes a `dogont:OnStateValue` and a `dogont:OffStateValue`). The current state of a device is therefore defined by a list containing one `dogont:StateValue` per each `dogont:State`.

Consider a shutter actuator model as an example (Figure 3.4) for states modeling. The shutter actuator is represented in DogOnt as having one `dogont:ShutterState` that, in turn is related to five state values: `UpStateValue`, `DownStateValue`, `LoweringStateValue`, `RaisingStateValue` and `RestStateValue`. These values represent the visible operating conditions of the actuator.

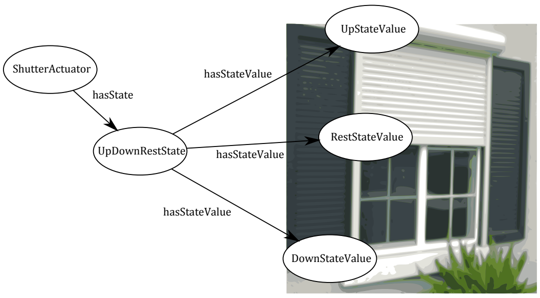


Figure 3.4. State modeling of a roller shutter actuator

3.2 Dog

To effectively operate, a SmE requires a software component named *middleware* (or gateway) that represents the "intelligent" part of the system. This component is typically responsible of merging the flow of data coming from the devices through their diverse networks and make them available to other software components present in the system, and it uses some AI techniques for taking decisions or supporting the environment inhabitants. Moreover, an important task assigned to middleware is to facilitate interoperability, i.e., to help devices and networks created by different providers and speaking different protocols to effectively understand each other and converge into a unique and uniform representation that can be easily understood by other software in the SmE [2]. This is particularly true if we consider devices and appliances present in a home automation system.

Dog (formerly known as Domotic OSGi Gateway) [12] is an ontology-powered middleware that exploits the OSGi framework[3] as a coordinator for supporting dynamic module activation, hot-plugging of new components and reaction to module failures. Such basic features are integrated with DogOnt, for supporting the integration of different networks, to implement inter-network automation scenarios, to support logic-based intelligence, and to access devices and appliances through an interface based on a neutral representation. Moreover, cost and flexibility concerns take a significant part in the platform design: Dog is an open source solution capable of running on low cost (and low performance) hardware such as a Raspberry Pi.

Dog, born in 2008 with the name of Domotic OSGi Gateway, is actively maintained and developed by the e-Lite research group at the Politecnico di Torino. During 2013, Dog has been improved in its design and underwent a code refactoring: my contribution was on the "core" and "communication" parts, to improve overall performances and provide a better compliance with the OSGi specifications. This refactoring led to Dog 3.0, released with the Apache License 2.0 on GitHub[4].

3.2.1 Architecture

Dog design principles include versatility, addressed through the adoption of an OSGi based architecture, advanced intelligence support, tackled by formally modeling the environment and by defining suitable reasoning mechanisms, and accessibility to external applications through a well defined, standard API available in REST (over HTTP) and via WebSocket. OSGi is a Universal Middleware that provides a service-oriented, component-based environment for developers and offers standardized ways

[3] `http://www.osgi.org/`, last visited on January 2014

[4] see also the Dog website at `http://dog-gateway.github.io` (last visited on February 2014)

to manage the software life cycle, as well as remote management. It provides a general-purpose, secure, and managed framework that supports the deployment of extensible services known as *bundles*.

Dog is organized in a layered architecture with **4 layers**, each dealing with different tasks and goals, ranging from low-level interconnection issues to high-level modeling and interfacing (Figure 3.5). Each layer includes several OSGi bundles, corresponding to the functional modules of the system.

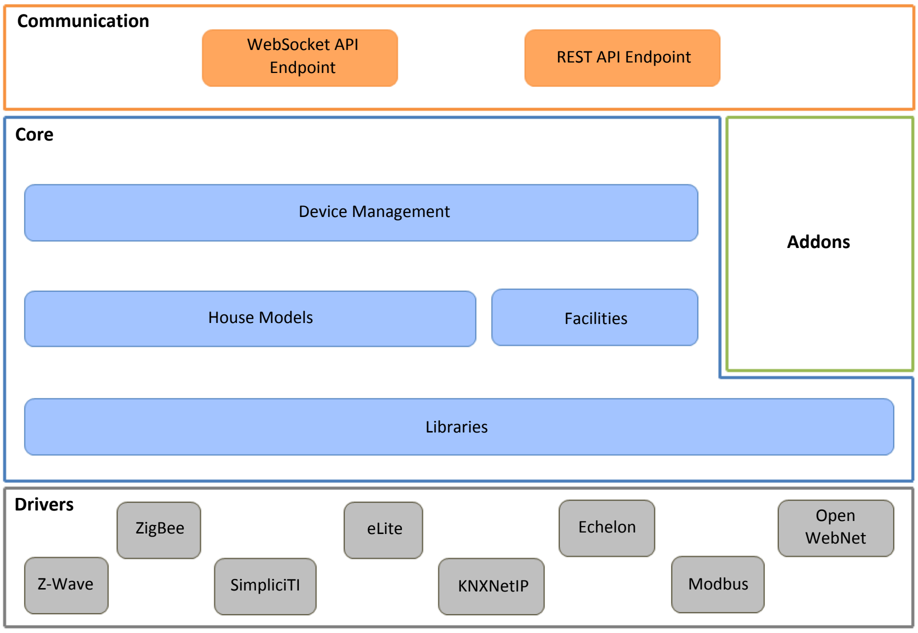


Figure 3.5. Dog Architecture

The first layer, **Drivers**, encompasses the Dog bundles that provide an interface to the various home and building automation networks to which Dog can be connected. Each network technology is managed by a set of dedicated drivers, which abstract network-specific protocols into a common, high-level representation that allows to uniformly drive different devices.

The second layer, **Core**, hosts the core intelligence of Dog, based on the DogOnt ontology, that is implemented in the Semantic House Model bundle. Moreover, it provides a set of common libraries and services useful to the entire systems or expected from the OSGi specifications.

The third layer, **Addons**, includes additional bundles for injecting further capabilities or more intelligence to the "core" part of the system, such as data storage, stream processing, rule engine, etc.

Finally, the fourth layer, **Communication**, provides the bundles offering access to external applications, either by means of a REST Endpoint or via WebSocket.

In the following, relevant services and functionalities of each layer are described in more detail.

Drivers Layer

In order to interface home and building automation networks, Dog provides a set of Drivers, one per each different technology (e.g., KNX NetIP, Z-Wave, etc.). Drivers implementation and operation follow the OSGi Compendium Specification [48]. Each Driver implements a "self-configuration" phase, in which it interacts with the OSGi framework to retrieve all the needed low-level information, according to the specific technology, e.g., the device address(es) or its ID in the network. For each technology, a *Network Driver*, a *Gateway Driver* and at least one *Device Driver* must exist.

The **Network Driver** handles network-level communication, in terms of protocols, connections and polling (when needed). It defines also the network access APIs for all the Device Driver bundles of the same technology. In Dog, only one Network Driver may exist for each technology.

The **Gateway Driver** supports multi-gateway operation for a given technology and handles the association between devices and their gateways. It permits to install Device Driver bundles if and only if the corresponding network gateway is present at the configuration level. Moreover, according to the specific technology, it can provide gateway-specific commands and functionalities. In Dog, only one Gateway Driver may exist for each technology.

A **Device Driver** implements the DogOnt device features for a given class of devices, i.e., it translates ontology-defined commands, functionalities and states into network level messages. Typically, in Dog, a Device Driver is created for each DogOnt device class.

Currently, eight different technologies are supported by Dog, each with a different number of device categories covered: KNX NetIP, Modbus (RTU and TCP), Echelon iLon100, eLite (i.e., a set of simulated drivers), BTicino OpenWebNet, Z-Wave, ZigBee Home Automation profile, and SimpliciTI.

Core Layer

The *Core Layer* encompasses the core intelligence of Dog and provides a set of common libraries and services useful to the entire system or expected from the OSGi specifications.

The **Device Management** category comprises three bundles for handling all the life-cycle of a Device and its status variables.

The *Device Factory* bundle is responsible to create and destroy Device instances in the framework, according to the runtime configuration it receives. Such a configuration can be provided by one of the House Models or, optionally, injected by a Gateway Driver, if the underlying network supports the runtime discovery of new devices.

The *Device Manager* bundle implements the OSGi Device Access Specification [48]: it manages the procedure for matching and attaching a Device to the "right" Driver (if any), each time one of them is added, modified or removed in the framework.

The *Monitor Admin* bundle implements the OSGi Monitor Admin Service Specification: it provides unified access to any declared status variables defined in the framework. Moreover, it offers security checking and scheduling of periodic or event-based monitoring jobs, i.e., it sends events related to some specified status variables according to defined rules.

The **House Models** category encompass two bundles: the Semantic and the Simple House Model.

The *Semantic House Model* manages the building (or home) description in form of DogOnt instances, thus supporting model merging, implementing classification and basic reasoning, supporting the extraction of interoperation rules and, potentially, providing access to all the properties defined in the ontology. Moreover, it can generate the XML configuration to be used by the Simple House Model.

The *Simple House Model* manages the building description in XML and it is typically used for Dog instances targeted to devices with low computational power. It does not have any model merging capabilities or reasoning support.

The **Facilities** category provides two bundles: a *Logger* and a *Clock*. Such bundles offer, respectively, a console logger through the OSGi LogService, and an internal clock service that triggers a time event each second.

The **Libraries** category consists of six bundles that act as repositories of classes, interfaces, and various services needed by the other Dog bundles. The *JAXB Library* provides XML serialization and de-serialization for handling the Simple House Model configuration and for some messages provided by the Communication Layer. The *Measure Library* provides unit of measures and related operations by offering the JScience library to all Dog bundles; it defines also some unit of measured unsupported by JScience. The *Semantic Library* encapsulates and makes available all semantic-related libraries, such as Apache Jena, Pellet and a SPARQL query facilitator. The *Stream Library* define some types of events to offer a unified access to them for a uniform handling by the Complex Event Processor that can be used in an Addons bundle. The *org.rxtx*, similarly to the Measure Library, exports the serial port API library (RXTX) to all Dog bundles.

Eventually, the *Core Library* is the most important:

- it contains all the possible devices, functionalities, states and state values as defined in DogOnt; all these classes and interfaces are programmatically generated from DogOnt, thus ensuring a formal and full compliance with the ontology representations.
- it encompasses all the possible device implementations, defined as abstract classes and programmatically generated from the previous part.
- it provides core-level notifications, i.e., not defined in DogOnt, as well as common data structures needed by the other Dog bundles.
- it provides utility classes to other bundles, to avoid code duplications and repetitions.

Addons Layer

This layer provides additional bundles for injecting further capabilities or more intelligence to the previous layer. Currently, it comprises the following bundles.

The *Rule Engine* bundle provides a rule engine runtime for defining automation scenarios, interoperation and complex device behaviors; it is programmable through dedicated XML messages, coming from the Communication layer or read from disk. It uses notifications as triggers, states as constraints, and commands as rule consequent actions.

The *Power Bundle* offers power consumption estimation based on actual measures, typical or nominal values defined in the DogOnt power extension and exploited by the *Power Model*. This model provides power-specific query functionalities and plugs in the Semantic House Model. Both the DogOnt power extension and the Power Bundle will be better detailed in Chapter 8.

The *Stream Processor* bundle provides stream processing capabilities by handling measure and boolean events for processing them in the spChains framework [49].

Finally, the *Event Storage* bundle maintains a storage of relevant events (e.g., measures) by using a in-memory database with a small footprint. Memorized data can be then visualized and analyzed according to any specific need.

Communication Layer

The *Communication Layer* provides access from external, non-OSGi, applications by offering two alternative endpoints: a REST API and a WebSocket API.

Both APIs use the same basic message structures and expose the same functionalities and information: they help retrieving the building configuration, sending commands to devices managed by Dog, handling the building structural information (rooms, flats, etc.), getting the devices status, etc. The only exception concerns asynchronous events (i.e., notifications) that may come from any device handled by Dog: they are provided only by the WebSocket endpoint due to the synchronous nature of HTTP.

3.2.2 Implementation

Dog has been implemented in Java, as a set of 88 self-developed OSGi bundles running on the Equinox OSGi implementation. Adjunctive but mandatory bundles, implementing standard OSGi services, are taken from diverse OSGi open source implementation and mainly from Equinox and Apache Felix.

The DogOnt ontology is managed by the Semantic House Model using the Apache Jena API, while the external API modules exploit the JAXB (Java Architecture for XML Binding) project for handling XML contents and the Jackson JSON Processor for managing JSON documents. The REST endpoint uses the Jersey RESTful Web Services framework that provides a good toolkit for developing RESTful Web Services in Java.

The current version of Dog (3.0) is released on GitHub, in both binary and source formats (see `http://github.com/dog-gateway`), under the Apache License 2.0, while previous versions of the gateway were released on SourceForge[5]. Dog runs on very cheap computers such as the Raspberry Pi: a credit-card sized computer with an ARM processor at 700MHz and 512 MB of RAM.

3.3 Conclusions

This chapter introduced the "founding technologies" needed for the supporting the applications and the systems presented in the rest of this thesis, i.e., the latest version of the Dog gateway and the DogOnt ontology. The chapter showed the capability of Dog and the different type of usage of DogOnt in the gateway itself.

Building upon the current and previous versions of the Dog gateway and, when needed, extending the Dog capabilities and the "smartness" provided by DogOnt, the following chapters present various approaches aimed at improving the interaction with a smart home empowered by technologies that are comparable to Dog and DogOnt.

[5] `http://sourceforge.net/projects/domoticdog/`, last visited on January 2014

Chapter 4

DOGeye: Gaze-based Home Interaction

This chapter describes the design and development of DOGeye, one of the first home control applications explicitly designed for gaze-based interaction, taking into account the guidelines proposed by COGAIN, an European Network of Excellence. DOGeye is a multimodal eye-based application for the management and the control of a Smart Home, based on state-of-the-art technologies in both eye tracking systems and SmE. It enables people to control their homes through different input devices, possibly combined, so that it does not limit itself to eye tracking only. The presence of various input modalities allows the application use by other people present in the house and offers different alternatives to the persons affected by severe and possibly evolving impairments, such as ALS (Amyothrophic Lateral Sclerosis).

4.1 Motivation

In the last 10 years, (smart) home automation gained a new momentum, thanks to an increased availability of commercial solutions (e.g., X10 or Z-Wave) and to steadily reducing costs. The evergreen appeal of automated, "intelligent" homes together with a raising technology maturity has fostered new research challenges and opportunities in the field of "smart" environments. According to the Mark Weiser definition, a Smart Home system, that in this chapter we decline as a home automation or environmental control system[1], is "a physical world that is richly and invisibly interwoven with sensors, actuators, displays and computational elements, embedded seamlessly in the everyday object of our lives, and connected through

[1] we only consider systems currently available on the market such as X10, KNX, Z-Wave and ZigBee HA

a continuous network" [3], providing ways for controlling, interacting and monitoring the house. The idea behind this vision is that homes of tomorrow would be smart enough to control themselves, understand contexts in which they operate and perform suitable actions under inhabitants' supervision [50]. Although smart and autonomous homes might raise controversial opinions on how smart are they or should they be, currently available commercial solutions can start playing a relevant role as enabling technology for improving the care of the elderly [51, 52] and of people with disabilities [53, 54], reducing their daily workload in the house, and enabling them to live more autonomously and with a better quality of life. Even if such systems are far from cutting-edge research solutions, they are still really complex to master since they handle and coordinate several devices and appliances with different functionalities and with different control granularities.

In particular, among other disabilities, people who have severely impaired motor abilities can take great advantages from eye tracking systems to control their homes, since they generally retain normal control of their eyes, that become therefore their preferential stream of interaction [55]. Eye tracking can transforms such a limited ability into both a communication channel and an interaction medium, opening possibilities for computer-based communication and control solutions [56]. Even if eye tracking is often used for registering eye movements in usability studies, it can be successfully exploited as alternative input modality to control user interfaces. Home automation can then bridge the gap between software and tangible objects, enabling people with motor disabilities to effectively and physically engage with their surroundings [57]. Several house control interfaces have been proposed in the literature, i.e., applications to allows users to control different types of devices in their homes, to handle triggered alarms, etc. Such interfaces, either based on conventional unimodal [58] or multimodal interactions [17] (e.g., mouse, remote controller, etc.), are too often uncomfortable and/or useless for people with severe impaired motor abilities, and only few of them have been specifically designed and developed to be controlled with eye movements.

In 2004, applications based on gaze interaction have been analyzed by a European Network of Excellence, named COGAIN (*Communication by Gaze Interaction*), to evaluate the state-of-the-art and to identify potential weaknesses and future developments. According to the report "D2.4 A survey of Existing 'de-facto' Standards and Systems of Environmental Control" [59], the COGAIN Network identified different problems in eye-based house control applications, such as the lack of advanced functionalities for controlling some appliances of the house, the absence of interoperability between different smart house systems or the difficulty to use an eye tracker for realizing some actions. In a subsequent report [13], COGAIN members proposed solutions to overcome the discovered problems. In particular, they proposed 21 guidelines to promote safety and accessibility in eye tracking based environmental control applications.

4.2 Eye Tracking Basics

To better understand the principles and implementation of eye controlled interface, this section defines some terms and features pertaining to eye movements and eye tracking.

The eye does not generally move smoothly over the visual field; instead, it makes a series of quick jumps, called *saccades*, along with other specialized movements [60]. A saccade lasts 30 to 120 ms, and typically covers 15 to 20 degrees of visual angle [61]. Between saccades, the *gazepoint*, i.e., the point in a scene where a person is looking, stays at the same location (with a slightly tremor) for a *fixation* that lasts from 100 to 400 ms; a longer fixation is called *dwell* [55].

Eye positions and their movement relative to the head can be measured by using different methods, such as computer vision techniques. One of these techniques is the so-called *Corneal Reflection* technique that consists in sending a small infrared beam toward the center of the pupil and estimating the changes in its reflexion (*eye tracking*). Eye tracking has several distinguishing features [61]:

- it is *faster* than other input media, as Ware and Mikaelian [62] observed; in fact, before the user operates any mechanical pointing device, she usually looks at the destination to which she wishes to move;

- it is *easy* to operate, since no training or particular coordination is required to look at an object;

- it shows where is located the user's *focus of attention*; an eye tracker input could be interpreted as an indication of what the user points at, but it can also be interpreted as an indication of what the user is currently paying attention to, without any explicit input action on her part;

- it suffers from *Midas Touch* problem: the user expects to be able to look at an item without having the look cause an action to occur. This problem can be overcome by using techniques such as dwell time or blink selection;

- it is *always on*; in fact, there is no natural way to indicate when to engage the input device, as there is with grasping or releasing the mouse;

- it is *noninvasive*, since the observed point is found without physical contact;

- it reduces *fatigue*; if the user uses an eye tracker input instead of other manual pointing devices, movements of arms and hands will be reduced and will cause less fatigue;

- it is *less accurate* than other pointing devices, such as a mouse.

Because of these features, an eye tracking based interface has some specific peculiarities: for example, graphical widgets and objects may be bigger than in traditional user interfaces, due to eye tracking lower accuracy; the pointer is often absent, since its presence could divert users' attention [63], but it is replaced by other forms of visual feedback.

To overcome the "Midas Touch" problem, many interfaces use the *dwell time* technique. By using such a technique, the user can select a widget present on a user interface only if she continues to look at it for a sufficiently long time. The amount of time is, generally, customizable by the user itself.

Moreover, interaction with eye-based interfaces can be improved by exploiting the *Selection-Action strategy* (SA), already used in the iAble application[2] and whose basic principle was proposed by Razzak et al. [64]. This strategy permits to separate the selection of an object from the activation of its associated actions. The *selection* is the process of choosing an object and displaying its related options, while the *action* permits to perform some tasks on the selected object. The selection-action strategy is generally implemented by showing two separate areas to interact with: one is used only for selection, with a really short dwell time; the other is used for actions, with a longer dwell time, controllable by users. Two interaction patterns lie at the basis of SA: the *non-command based interaction*, used for selection, and the *command based interaction*, used for actions. In the *non-command based interaction* pattern, the computer observes and interprets user actions instead of waiting for explicit commands. By using this pattern, interactions become more natural and easier, as indicated by the work of Tanriverdi and Jacob [65]. In *command based interactions*, instead, the user explicitly directs the computer to perform some operations.

4.3 The COGAIN Guidelines

The COGAIN (*Communication by Gaze Interaction*) project was launched in September 2004, as a Network of Excellence supported by the European Commission's Information Society Technology under the 6th framework programme, with the goal of "integrating cutting-edge expertise on gaze-based interface technologies for the benefit of users with disabilities." The project gathered over 100 researchers belonging to the world's cutting-edge research groups and companies with leading expertise in eye tracking integration with computers and in assistive technologies for people with motor impairments. COGAIN also involved the advice of people coming from hospitals and hospices, working daily with persons with motor impairments. Thanks to

[2] a SRLabs commercial software

the integration of research activities, the network developed new technologies and systems, improved existing gaze-based interaction techniques, and facilitated the implementation of systems for everyday communication (for more information see the COGAIN web site[3]).

COGAIN considered home automation and smart homes as an opportunity for eye tracker users to live in an autonomous way. For this reason, in 2007, the COGAIN project published a Draft Recommendations for Gaze Based Environmental Control [13]. This document proposes a set of guidelines for developing home control interfaces based on eye interaction. The guidelines originated from a set of realistic use case examples, describing typical actions that a user with impairments can do in her smart environment, and underwent an evaluation and validation process in the COGAIN project.

Gaze Based Environmental Control guidelines (see Table 4.1) are grouped in 4 main categories:

1. *Control applications safety*: guidelines concerning the behavior of the application in critical conditions, such as alarms and emergencies.
2. *Input methods for control application*: guidelines about input methods that the control applications should support.
3. *Control applications significant features*: guidelines impacting the management of commands and events within the house.
4. *Control applications usability*: guidelines concerning the graphical user interface and the interaction patterns of the control applications.

Each guideline is associated to a priority level (PL), following the typical W3C style:

- Priority Level 1: the guideline MUST be implemented by the applications, since it relates to safety and basic features;
- Priority Level 2: the guideline SHOULD be implemented by the applications.

Control interfaces for smart environments must face three main issues that lie at the basis of most guidelines:

Asynchronous control sources The control interface is not the sole source of commands: other house occupants may choose to operate on wall-mounted switches, some external events may change the status of some sensors, etc.

[3] `http://www.cogain.org/`, last visited on January, 2014

Guideline	Content	PL
1.1	Provide a fast, easy to understand and multi-modal alarm notification	1
1.2	Provide the user only few clear options to handle alarm events	2
1.3	Provide a default safety action to overcome an alarm event	1
1.4	Provide a confirmation request for critical & possibly dangerous operations	1
1.5	Provide a STOP Functionality that interrupts any operation	1
2.1	Provide a connection with the COGAIN ETU-Driver	1
2.2	Support several input methods	2
2.3	Provide re-configurable layouts	2
2.4	Support more input methods at the same time	2
2.5	Manage the loss of input control by providing automated default actions	2
3.1	Respond to environment control events and commands at the right time.	1
3.2	Manage events with different time critical priority	1
3.3	Execute commands with different priority	1
3.4	Provide feedback when automated operations and commands are executing	2
3.5	Manage Scenarios	2
3.6	Communicate the current status of any device and appliance	2
4.1	Provide a clear visualization of what is happening in the house	1
4.2	Provide a graceful and intelligible interface	2
4.3	Provide a visualization of status and location of the house devices	2
4.4	Use colors, icons and text to highlight a change of status.	2
4.5	Provide an easy-to-learn selection method.	2

Table 4.1. COGAIN Guidelines

The control interface needs therefore to continuously update the status of the house, i.e., icons, menus and labels must timely change according to the home status evolution, to provide a coherent view of the environment *(Guidelines 3.6, 4.1)*.

Time-sensitive behavior In an alarm condition the user is normally put in a stressful condition: she has limited time to take important decisions, which may pose threats to her safety. In such stressful conditions, eye control may become unreliable or, in some cases, not functional. In this case the control interface must offer simple and clear options, easy to select, and must be able to take the safest action in case the user cannot answer in time *(Guidelines 1.1, 1.2, 1.3, 1.4, 3.2, 3.3)*. Time-sensitive behaviors include automated actions, initiated by rules (e.g., closing the windows when it is raining). In this case the user should be allowed to interrupt any automatic action or to override it at any time. The control interface should make the user aware that an automatic action has been started, and offer ways to interrupt it *(Guidelines 1.5, 3.5).*

Structural and functional views Control interfaces can organize the home information according to two main logics: structural and functional. Most environmental control applications apply the structural logic and display information mimicking the physical organization of devices in the home. This choice, however, cannot address global or "not-localized" actions as switching the anti-theft system on, or set the temperature of the house. Functional logic, instead, is best suited for tackling not-localized options and to support type-driven interaction with interface elements, i.e., interaction involving actions having the same nature. Effective interfaces should find a good trade-off between the two logics *(Guidelines 4.1, 4.2).*

4.4 Logic Architecture

To overcome the shortcomings of currently available solutions and to provide a first reference home control application designed explicitly for supporting COGAIN guidelines through multimodal interaction, with a strong focus on eye tracking technologies, we designed, implemented and evaluated DOGeye. Following subsections describe in detail the logic architecture of gaze-based home interactions supported by DOGeye and provide useful insights on the application design and functionalities. DOGeye has been designed according to the *user centered design* approach [66] to be a COGAIN-compliant, multimodal eye-based application for controlling, interacting and monitoring a house. Interaction with automated (smart) homes is provided by *Dog* [12], an ontology-based gateway able to integrate and abstract functionalities of heterogeneous systems, thus offering a uniform, high-level access to home technologies.

Figure 4.1 shows the logic architecture of home control through gaze. DOGeye communicates with the smart environment exploiting Dog (on the right), thanks to a XML-RPC connection that allows exchanging all the needed information about the home, and directly controlling the available devices. The DOGeye connection with Dog allows to respond almost instantly (≈100ms) to environmental control events and commands, thus complying to the COGAIN Guideline 3.1.

DOGeye can either be controlled by gaze (main interaction channel), by communicating with the eye tracker through a universal driver named ETU-Driver [67] (Guideline 2.1), or can be used for exploiting other interaction mediums such as touch screens or traditional keyboards and mouses (not covered in this chapter),

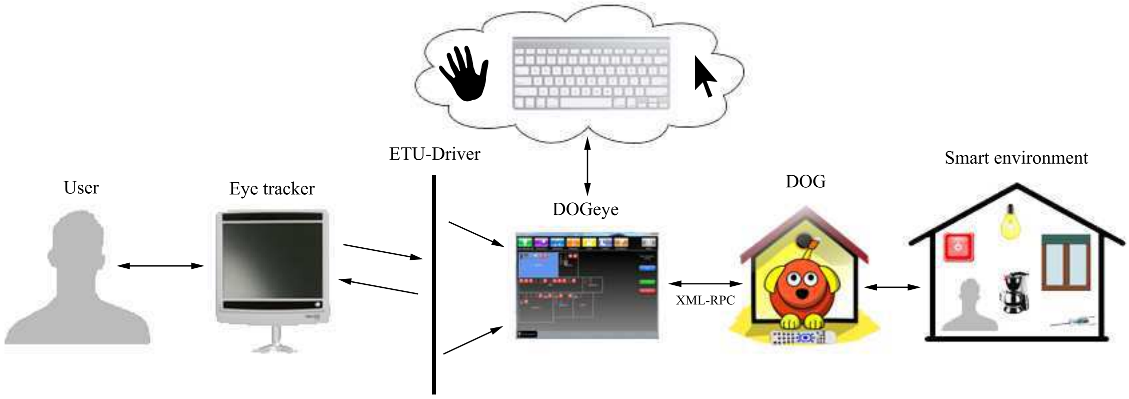


Figure 4.1. The context where DOGeye is inserted

thus fulfilling Guideline 2.2: *Support several inputs methods*. These input methods, moreover, are usable at the same time (according to *Guideline 2.4*), allowing the user to manage a possible loss of eye tracking input by using other interaction methods and giving a preliminary implementation to Guideline 2.5, i.e., manage the loss of input control by providing automated default actions.

4.5 Design

The DOGeye interface has been designed following an incremental specification paradigm where a first layout skeleton (see Figure 4.2) has been incrementally refined to explicitly comply with most of the COGAIN guidelines about environment control.

4.5.1 Design Rationale

According to the COGAIN guideline 2.3, the draft specification accounts for reconfigurable layouts appropriate for different eye tracking resolutions and precisions, even if the current DOGeye implementation is not yet complete in this regard. As suggested by guideline 4.2 on "graceful and intelligible interfaces", we adopted a color policy shared by all the interface elements, in a consistent manner, as shown in Table 4.2.

The interface is divided in four main logical areas:

- **Tabbed area** - in the upper part of the interface, it represents the *functional view* of the house. It contains tabs showing different views of the ambient, according to the type of the device to show. This area behaves according to the "selection" part of the Selection-Action pattern.

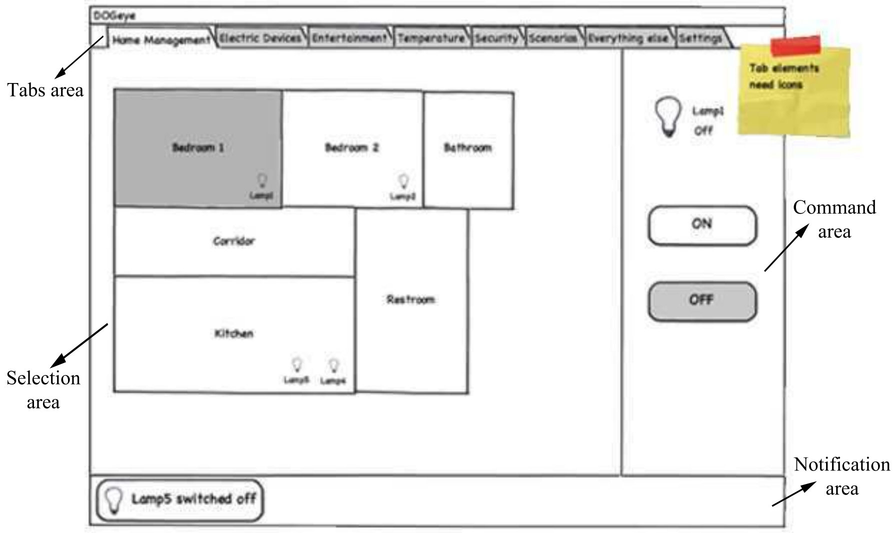


Figure 4.2. DOGeye Abstract Layout

Function	Description	Color
Positive behavior	Commands with a positive meaning, such as “open”	Green
Negative behavior	Commands with a negative meaning, such as “close” or “stop”	Red
Neutral behaviour	Commands with neither a negative or positive meaning, such as “set to. . . ”	Gray
Selection	Buttons for enabling/disabling selection functionalities	Black
House navigation	Commands for navigating within the house, such as “enter this room”	Blue

Table 4.2. Descriptions of the colors used for DOGeye buttons

- **Selection area** - in the left part of the interface, this large area represents the *structural view* of the house. It contains the house rooms and its devices and it allows selecting a room, device or a group thereof. Also this area behaves according to the “selection” part of the Selection-Action pattern.

- **Command area** - in the right part of the interface, it shows the commands supported by the object selected in the selection area. This area behaves according to the “action” part of the Selection-Action pattern.

- **Notification area** - located at the bottom of the interface, it shows notifications and alarms to the user.

The first two areas are designed for specifically addressing COGAIN requirements asking for a proper balance between functional and structural home views. In the functional view, devices are grouped according to their nature and functionality, rather than by location. The structural view, instead, follows the physical organization of the devices in the house, i.e., each device is located inside its containing room. The joint adoption of the two views defines two distinct navigation hierarchies: a functional hierarchy that lets users choose the device type or the kind of operation to accomplish, and a structural hierarchy, which allows users to choose specific devices inside the home.

The *Command area* fulfills Guideline 1.4: *Provide a confirmation request for critical and possibly dangerous operations.* DOGeye never acts on the basis of just one fixation, but always requires at least two: the first with a short dwell time and the second with a longer one. In fact, by using a short dwell time the user can easily select an object, but errors are possible: no harms, since the selection has been quick and the operation is simply reversible.

The last area - the *Notification area* - provides a feedback each time an operation or a command is executed in the house. It communicates also the current state of any device and appliance, thus implementing Guidelines 3.4, 3.6 and 4.1. Every status change of the house devices is notified in different ways, both visually and phonetically. For example, when the lamp in the kitchen is switched on, its icon will represent a lighted bulb, a notification with the lamp image and carrying the label "*The lamp in the kitchen is switched on*" will show up in the *Notification area* and the speech system, built in the Windows OS, speaks the same sentence reported in the notification. Moreover, the speech system gives the user a feedback for every actions she does, e.g., it says "You are in the kitchen" when the user "enters" the kitchen using the application. In this way, DOGeye actuates the Guideline 4.4: *Use colors, icons and text to highlight a change of status.*

4.5.2 DOGeye User Interface

The final appearance of DOGeye is presented in Figure 4.3; it is possible to notice eight tabs in the *Tabbed area*, each with a different function. Every tab has a different icon with a different color and an explanatory text: associating an icon with a text label significantly reduces the possibility of misinterpretation and error by a user, compared with the use of only an icon or just a short text [68].

The eight tabs with their functions are:

- **Home Management** contains the "basic" devices present in a house, i.e., devices belonging to the mains electricity wiring, such as shutters, doors, lamps, etc.

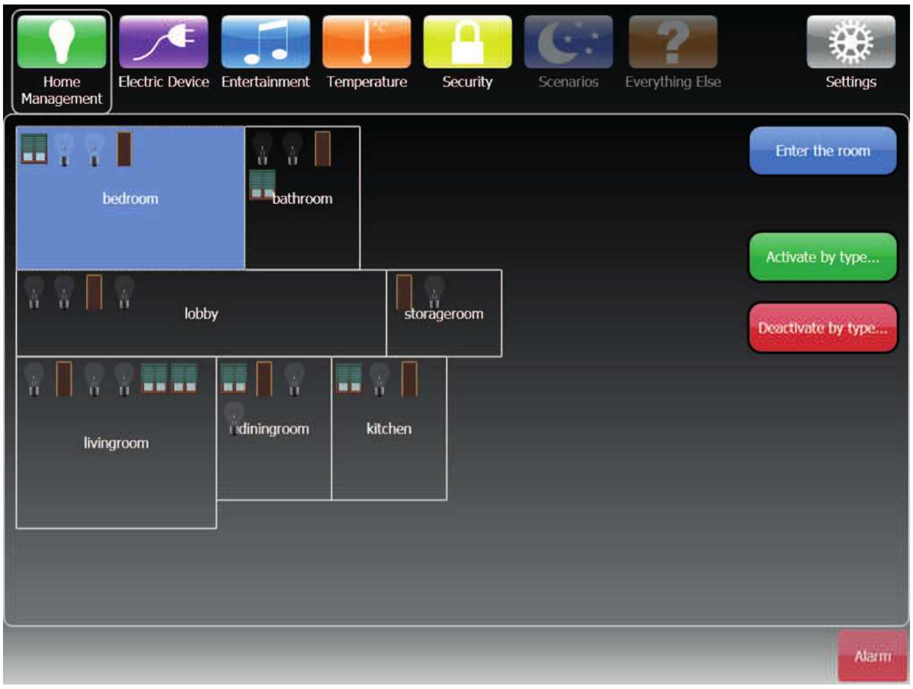


Figure 4.3. DOGeye User Interface

- **Electric Device** contains the electrical appliances not belonging to the entertainment system, such as a coffee maker.
- **Entertainment** contains the devices for entertainment, such as media centers and TVs.
- **Temperature** allows handling the heating and cooling system of the house.
- **Security** contains the alarm systems, anti-theft systems, etc.
- **Scenarios** handles the set of activities and rules defined for groups of devices, as suggested by Guideline 3.5.
- **Everything Else** contains devices not directly falling in the previous tabs, e.g., a weather station.
- **Settings** shows controls for starting, stopping and configuring the ETU-Driver.

All the tabs report the home plan (i.e., the map) of the house and the current state of devices, represented as a changing icon located in the room in which the device is positioned. This architectural view of home devices enables DOGeye to satisfy COGAIN Guidelines 4.1 and 4.3 regarding visualization of what happens in the house and where. Given a tab selection, e.g., “Home Management”, users can gather an overview of the current state of the house devices and may decide to turn on/off some of them. To actuate a specific device, users are required to “enter” the room containing the device and to command it through the Selection-Action interaction pattern.

Sample scenario Imagine that Sam, a man with severe mobility impairments wants to turn on the ceiling lamp in the kitchen. In the typical DOGeye interaction, he first chooses the “Home Management” tab by briefly looking at it (first level of selection), then he looks for a moment at the kitchen on the map (second level of selection), and then he fixates at the “Enter the room” button for a longer dwell time (action). After entering the room, Sam can see the list of all devices present in that room; he briefly looks at the ceiling lamp to select it. Then he gazes at the “On” button present in the Command area, to turn the lamp on. If, afterwords, Sam wants to prepare a warm coffee, he needs to switch to the “Electric Device” tab, using the same procedure.

Tabs are designed to act as isolated points of access to the home. This means that selections made in each tab are independent from each other and that the state of each tab representation is independent from all the others. With reference to Sam’s case, if Sam returns to the “Home Management” tab after switching on the coffee maker he will find the kitchen still selected and DOGeye will still be showing the devices in the kitchen and not the plan of the house, since he didn’t “leave” the kitchen before changing tab.

This feature is called *tab isolation*: each tab is independent from the others, so that a selection or an action made in one of them is always preserved. Next paragraphs better detail the most relevant tab functionalities.

4.5.3 Home Management and Electric Devices

These two tabs include the most common devices present in a house, e.g., power outlets, lamps, door actuators, window and shutter actuators, etc. These devices are controllable individually or in group. According to Guideline 4.5, i.e., *Provide an easy-to-learn selection method*, we implemented *single and multiple selection*, as summarized in Table 4.3.

Single selection is divided in *normal* and *implicit* selection. The simplest modality is *normal single selection*: by looking at an icon, the user selects the correspondent object. This selection acts both for selecting a room from the house map and for selecting a device or an appliance once inside a room.

Selection modality		**Activated**		
		where	*by looking at. . .*	*what happens?*
Single	normal	house level	a room	the chosen room is selected
		room level	a device	the chosen device is selected
	implicit	house level	a room with only one device	the only device present inside the room is selected
Multiple	normal	room level	"Multiple selection" button	it is possible to select more than one device
			"Select by type. . . " button	all the devices of a chosen type are selected
	implicit	house level	"Multiple selection" button	it is possible to implicitly select the devices present in more than one room
			"Activate/Deactivate by type. . . " buttons	all the devices of a chosen type, present in a room, are activated/deactivated

Table 4.3. Summary of the different selection modalities present in DOGeye

The *implicit single selection* occurs at the "house map level", when a selected room has only one device that is automatically (i.e., implicitly) selected by selecting the containing room: in this case, the Command area shows directly the commands for that device, as seen in Figure 4.4.

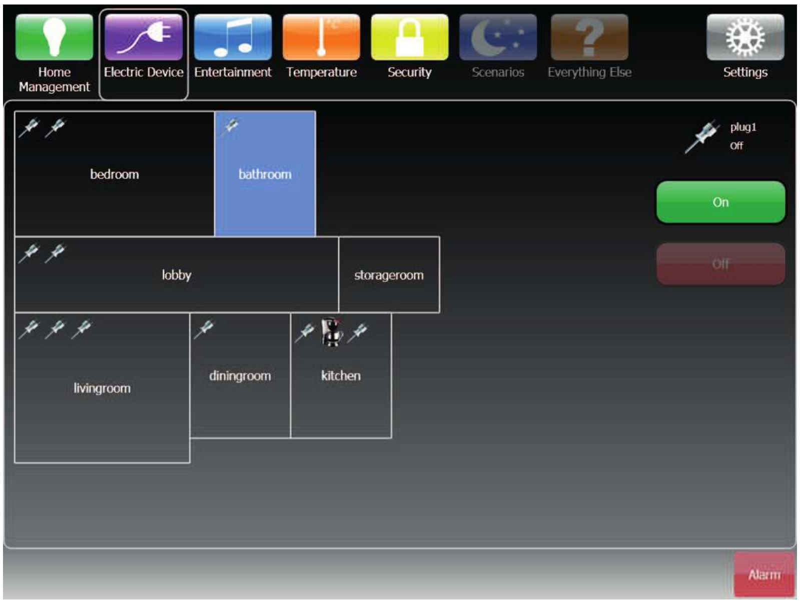
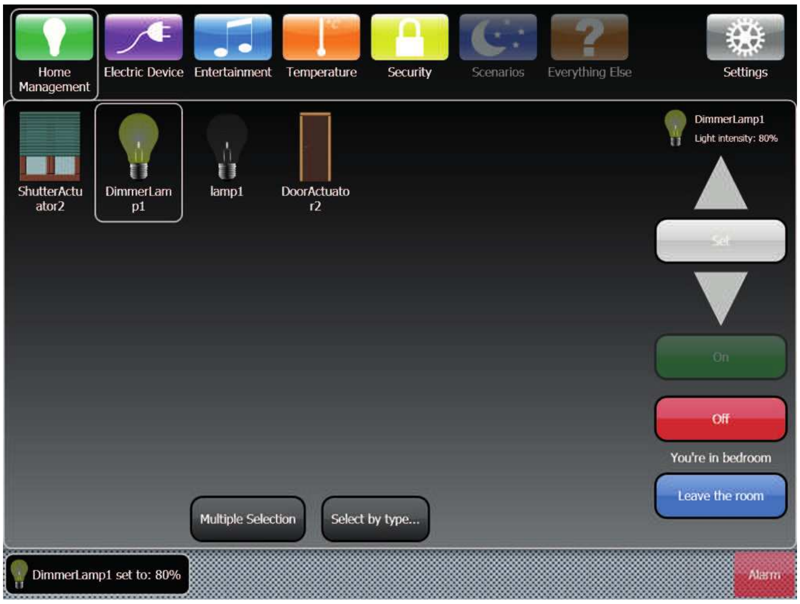


Figure 4.4. What happens when a room contains only one device

Multiple selection is also divided in *normal* and *implicit*. The *normal multiple selection* involves multiple devices present in a room. As shown in Figure 4.5, by looking at the multiple selection button, it is possible to select a subset of devices of the same type and then control them using one of the associated commands.



You're in bedroom

DimmerLamp1 set to: 80%

Figure 4.5. Example of devices inside a room in the Home Management tab

The *implicit multiple selection* occurs at the “house map level” and it is realized with two buttons, located in the Command area when a room is selected: “Activate by type. . . ” and “Deactivate by type. . . ”. When the user looks at one of these buttons, a popup window appears (Figure 4.6) and it is possible to give a basic command to all the devices of the chosen type in that room, without “entering” it. For example, by looking at “Activate by type. . . ” and then selecting Dimmer Lamp, it is possible to turn on all the dimmer lamps present in the selected room, but not to set their luminosity.

As a subcase of multiple selection, we provided the interface of a “select all” functionality. In these tabs, by looking at the “Select by type. . . ” button present inside the room, it is possible to select all the devices of the type chosen through a popup window, and then control them using one of the associated commands.

4.5.4 Temperature

The “Temperature” tab allows to control the temperature of a room driving the heaters/coolers present in that room. When a room is selected, it is possible to turn

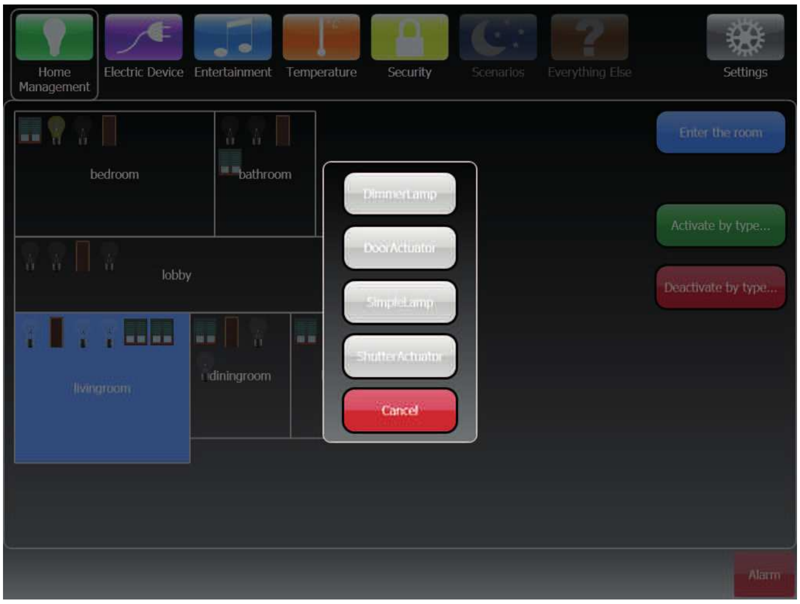


Figure 4.6. Example of implicit multiple selection in the Home Management tab

on/off the heating/cooling system inside the room and to set the room temperature in Celsius degrees.

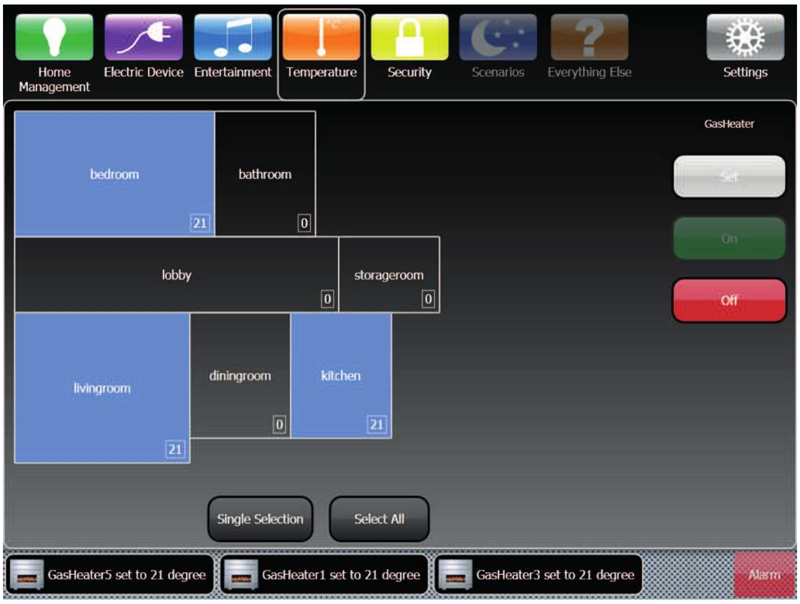
This tab only implements implicit *single* and *multiple selection*, due to the presence of only one device for each room. In this way, we allow to set a uniform temperature on different ambients. An example of the implicit multiple selection is shown in Figure 4.7.

In this tab, the "select all" functionality occurs at the "house map level" and it is realized with the "Select all" button: by looking at it, it is possible to select the whole house and so, implicitly, the whole heating/cooling system.

4.5.5 Security

The security tab allows to see what happens in any room provided with one or more cameras. Live videos or pictures can be viewed by entering the room containing the camera; for example, Figure 4.8 shows a room with one camera whose video can be accessed by looking at the upper part of the Command area.

The user may expand the video to full screen by simply looking at its icon. A button at the bottom (Figure 4.9) closes the video and returns to the previous view.



GasHeater5 set to 21 degree

GasHeater1 set to 21 degree

GasHeater3 set to 21 degree

Figure 4.7. Example of multiple selection in the Temperature tab

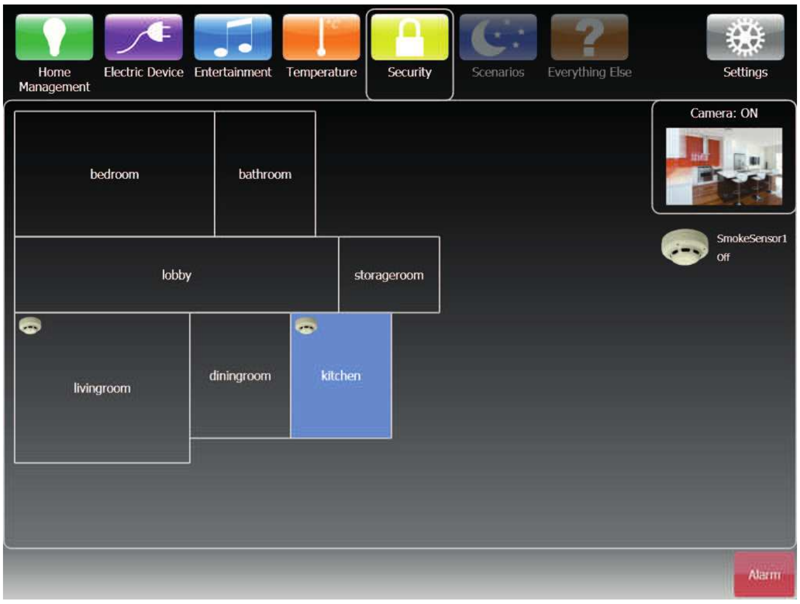


Figure 4.8. The Security tab

4.5.6 Asynchronous Alarm Events

To support handling of alarm events, as required by the COGAIN Guidelines 1.1 (*Provide a fast, easy to understand and multi-modal alarm notification*) and 1.2

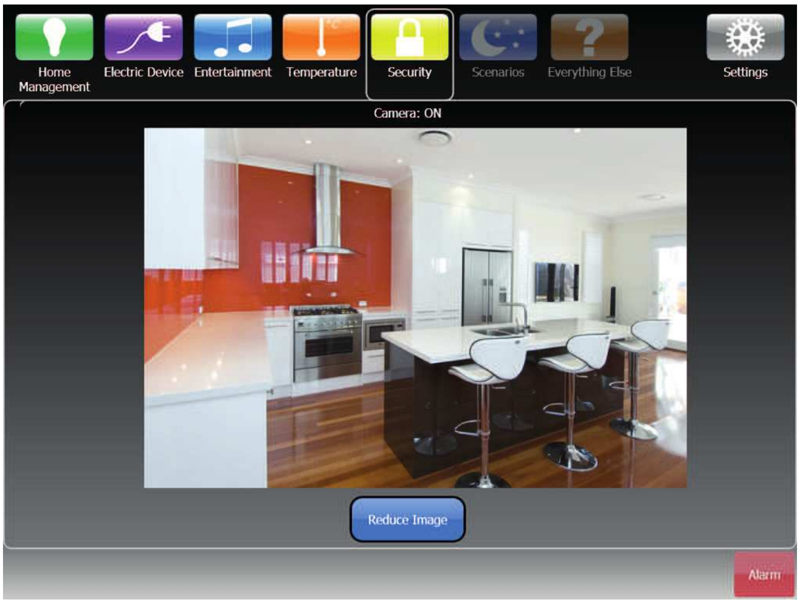


Figure 4.9. Full screen video in the Security tab

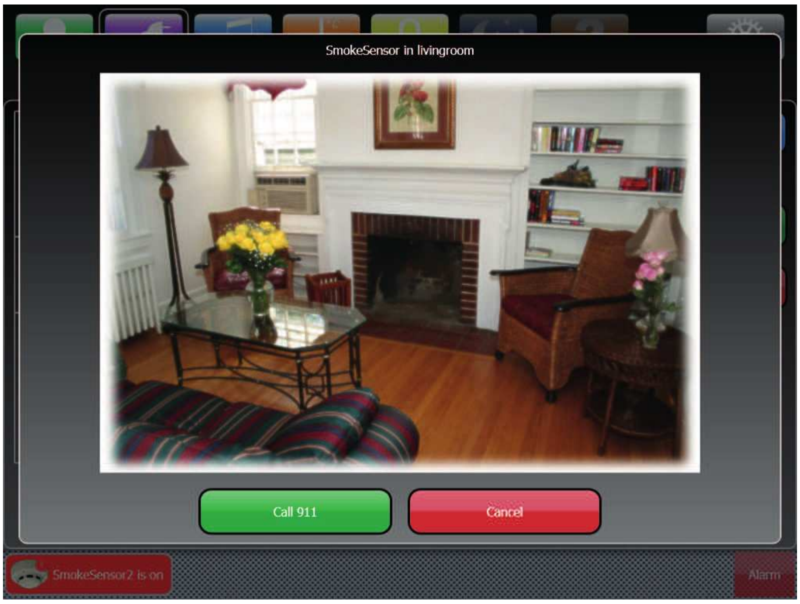
(*Provide the user only few clear options to handle alarm events*) we designed two alarm types: a *general alarm* and an *environmental alarm*. The *general alarm* functionality consists of a button, placed in the bottom right corner of the interface, that the user may use to draw attention and request help. When activated, this alarm generates an acoustic alert to attract attention, until the user stops it, by looking at a dedicated "STOP" button, thus complying with Guideline 1.5 that suggests such a functionality.

An *environmental alarm* is an event triggered if an alarm notification is received from Dog; it may occur in an asynchronous way respect to user actions. On the event activation, an acoustic alert is played with the purpose of attracting user attention.

As reported in Figure 4.10, the alarm event is managed by showing an overlay window containing a label which identifies the *device* that triggered the alarm and the *room* in which it is located, a *video* stream of what happens inside that room (if available) and *two buttons*, one for canceling the alarm and the other for handling the alarm in a safe way, for example by dialing 911 (or other emergency number).

If the user does not look at any button, DOGeye chooses for her the safest option after a pre-defined timeout (currently set to 20 seconds), e.g., "Call 911" (as required from Guideline 1.3). This could happen, for example, when the user is not able to look at any button due to a loss of the eye tracker calibration.

Alarm events are examples of activities with critical priority, that interrupt other user actions on her home. This behavior fulfills Guideline 3.2: *Manage events with different time critical priority* and Guideline 3.3: *Execute commands with different*



SmokeSensor in livingroom

SmokeSensor2 is on

Figure 4.10. An alarm generated by a smoke sensor

priority.

4.6 Interface Implementation

DOGeye is written in C#, using the *Windows Presentation Foundation* technology [69]. This solution allows to easily interface DOGeye with the ETU-Driver that is realized in C++ using COM Objects. It has a modular architecture shown in Figure 4.11 that also specifies the technologies adopted for the various modules and their communications. The main module is connected to external applications, i.e., Dog and the ETU-Driver.

The modular organization of DOGeye allows developers to easily edit the various parts of the program and possibly expand it to include new features.

The application includes a main window linked to *eight objects* representing the eight different tabs analyzed before. This main window also uses some configuration file and it is connected to *DOGleash* and an *eye tracking wrapper*. The DOGleash is the library used for the connection with Dog, while the eye tracking wrapper is the library managing the interoperation with the ETU-Driver. These two libraries are obviously linked with Dog and with the eye tracking driver, respectively.

DOGeye was tested on a *myTobii P10*, a remote and portable (but not wearable) eye tracker, running Windows XP and including a 15” single touch screen. Dog was locally installed on the eye tracker.

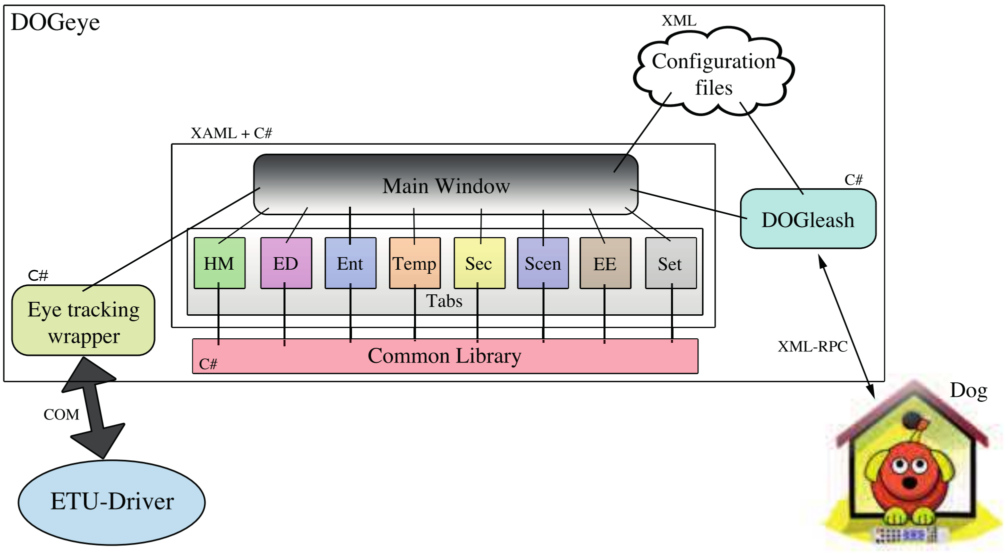


Figure 4.11. The general architecture of DOGeye

4.7 User Evaluation

The goal of the user evaluation is to identify the relative strengths and weaknesses of DOGeye, to roughly estimate the ability to use the interface without external hints and to check whether its advanced functionalities, such as multiple selection, are easy to discover and to use.

Eight participants used DOGeye in a controlled environment (described in 4.7.1) performing nine tasks (see Table 4.4) each, with Dog simulating the behavior of a realistic home through the DogSim capabilities [70]. Test tasks have been extrapolated from the COGAIN reference use cases and they have been tailored to the synthetic environment simulated by Dog; they reflect typical household tasks as also emerge from our design experience: we collaborated with the management staff of two smart building: the Maison Equipée in Val d'Aosta and the Don Gnocchi Foundation in Italy.

Participants never met during the evaluation. Their observations were used to allow qualitative analysis, to help identifying strong and weak points of the interface, and to identify future directions. Our analysis focuses on four basic questions to verify the usability of DOGeye:

1. How easily do users understand the tabbed organization? Are any icons hard to understand?

2. How easily and successfully do users find the options and tools they need?

3. How easily can the user control the smart home? What are the problems?
4. How easily can the user learn using the interface?

Eye control quality was not taken into account to study the usability of DOGeye, since we considered eye tracking simply as an input modality.

4.7.1 Methodology

We recruited 8 participants for our user study: 5 female and 3 male, aged 21 to 45 (with an average age of 31.3). Participants were selected for diverse skill level, especially about eye tracker usage. All except two worked in non-technology related fields, even if they use a computer more than four hours a day. Two groups of testers were selected: experienced (50%) and not experienced (50%) users. This design choice allows to gather some qualitative hint on the different needs DOGeye should be able to fulfill and on the degree to which such goal is reached. The study was held in Italian and the interface was localized accordingly.

A within-subject design was employed for both groups, where each subject in a group performed each task in counterbalanced order, to reduce order effects. We recorded a back video of the participant (Figure 4.12b), also capturing the screen content during the experiments.

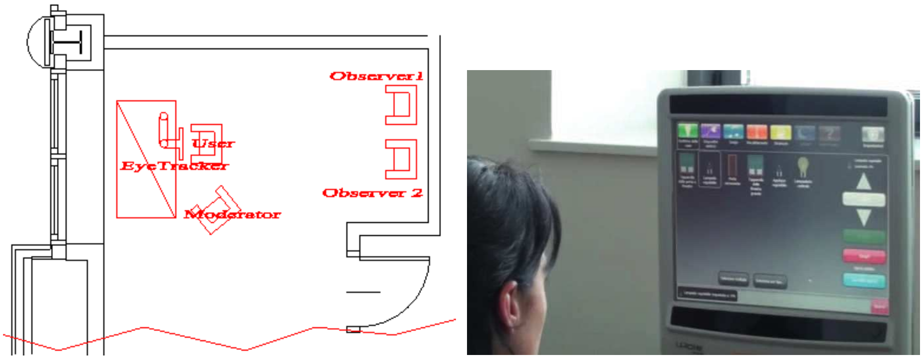
Controlled Environment Setup

Experiments were conducted into a controlled environment composed of a light-controlled room, inside which we positioned a desk carrying the eye tracker holding arm and the tracker itself. The eye tracker used in the study is the *myTobii P10* system and the adjustable holding arm allowed to set the tracker position to reach reasonable comfort of use for every study participant. The room used for the study hosted a moderator, seated near to the user, and two observers located in the background, not interfering with the test execution (typical *Simple Single-Room Setup*, see Figure 4.12). In general, methods followed recommendations for typical user studies [71].

Test Deployment

After a short introduction to the study and the collection of demographic data, a static DOGeye screenshot was shown to the participant to collect a first impression on the interface, by querying her agreement level (from “Strongly disagree” to “Strongly agree”) about these sentences:

1. I like the appearance of the program.



(a) Environment layout

(b) Environment setup

Figure 4.12. The controlled environment setup.

2. I think that the program is intuitive.
3. I think that the program layout is understandable.

It must be noticed that the previous sentences are translated from the original Italian formulation, therefore they just provide a mean for the reader to understand the posed questions while they do not retain the meaning nuances and the carefully formulated wording we used.

Warm Up Afterwards the eye tracker was calibrated to the user's eye and the participant was introduced to a simple game named "*Lines*" (Figure 4.13), already available on our eye tracker, to get her used to the eye tracking interaction. The goal of the game is simple: move some colored balls in order to make a line of five equal elements.

Task execution After two or three completed lines, DOGeye was started. We found that a reasonable value to set the *dwell time* of the eye tracker was 1.5 seconds, so we used that setting during all the study. This dwell time is longer than usual (i.e., 500 ms) and it explicitly aims at reducing as much as possible the eye tracking access gap for un-experienced users. Each user was told to complete a set of nine task (see Table 4.4), one at a time. For two of them, particularly simple, the participant was asked to use the *think-aloud* protocol, to verify her actions. Examples of proposed tasks include: "Turn on the lamp in the living room" and "If the heating system in the bedroom and in the kitchen is off, turn it on".

Test conclusion At the end, participants were given a questionnaire and asked to rate DOGeye in general, and to rate their agreement with the same sentences

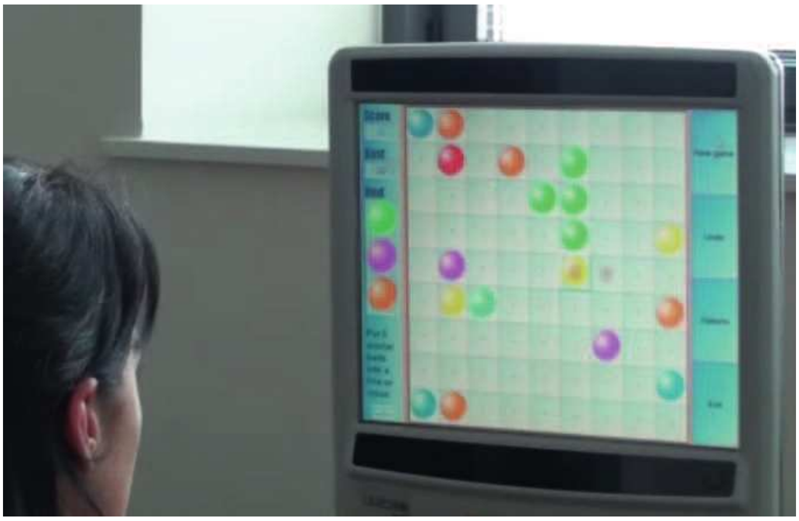


Figure 4.13. One participant uses Lines, during the study.

proposed earlier, just after seeing the screenshot of DOGeye. Users open comments or explanations were collected (e.g., problems found or explanation about something done during a task) through debriefing interviews. The duration of the entire experiment was dependent on eye tracker calibration problems and on how quickly participants answered the questions, but it ranged between 20 and 30 minutes.

Task	Description
T1	Turn on the lamp in the living room
T2	Plug the microwave oven in the kitchen
T3	Find a dimmer lamp, turn it on and set its luminosity to 10%
T4	Cancel the alarm triggered by the smoke detector
T5	Turn on all the lamps in the lobby
T6	If the heating system in the bedroom and in the kitchen is off, turn it on
T7	Set to 21 degree the heating system for the entire house
T8	Send a general alarm to draw attention in the house
T9	Read the smoke detector status and expand the video of the room to full screen

Table 4.4. The nine tasks used for the study

At the end of the study, we extrapolated from the videos the time (seconds) it took each participant to react to an alarm sent from the house, while the time it took participants to complete each task was not relevant due to the usage of the think-aloud protocol.

4.8 Results

We present and discuss quantitative as well as qualitative findings of our user study.

4.8.1 Quantitative Results

According to Nielsen's Alertbox[4] we calculate the *success rate* of each participant as the percentage of tasks that users complete correctly, also giving partial credit for partially completed tasks, i.e., those tasks completed with minor errors. We expected the participants with higher eye tracker experience to perform much better than others. Table 4.5 reports the success rates of the study, using the Nielsen's Alertbox notation, where the first four participants belong to the group of eye tracking "experts" (E) while the others are the "non experts" (NE). In the table, "S" indicates a successful task, "P" a partial success, and "F" a failed task.

As expected, "expert" participants had a mean success rate of 91.67% (standard deviation 8.33%) while non-experts reached a satisfying, yet lower, rate of 86.11% (standard deviation 8.33%), with an overall average of 88.89%. These success rates provide a general picture of how DOGeye supports users and suggest that only minor adjustments are needed to the interface design.

User	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**Success rate**
E1	*P*	S	S	S	*P*	S	*P*	S	S	83.33%
E2	S	*P*	S	S	*P*	S	*P*	S	S	83.33%
E3	S	S	S	S	S	S	S	S	S	100.00%
E4	S	S	S	S	S	S	S	S	S	100.00%
NE1	S	S	S	S	S	S	S	S	S	100.00%
NE2	S	S	**F**	S	*P*	S	S	S	S	83.33%
NE3	S	S	S	S	*P*	S	S	*P*	S	88.89%
NE4	**F**	**F**	*P*	S	S	S	S	S	S	72.22%

[a] *

E: expert-users, NE: non-expert users

Table 4.5. Success rate of the study

[4] `http://www.nngroup.com/articles/success-rate-the-simplest-usability-metric/`, last visited on January, 2014

The difference between the two user groups is clearly evident by looking at the time needed for replying to an alarm: the mean time for "experts" is 3.33 seconds (standard deviation 0.33 s) while the mean time for the others is 6.25 seconds (standard deviation 1.86 s), indicating that the difference of experience with the eye tracker favors the former.

4.8.2 Qualitative Results

The final questionnaire asked participants to give an overall grade to DOGeye, in a scale from 1 (the worst) up to 5 (the best). Results showed that they were satisfied of DOGeye performance, with a mean value of 4.25 (standard deviation 0.37).

In the test conclusion (see Table 4.6 for the results), participants were asked to express their agreement about four sentences of which the first three are equal to the preliminary questions described in Section 4.7.1:

1. I like the appearance of the program.

2. I think that the program is intuitive.

3. I think that the program layout is understandable.

4. It is easy to learn how to use the program.

Results from participants were satisfying: most users agree or strongly agree with the proposed sentences about DOGeye (see the bottom of Table 4.6 for details).

By comparing the results of the first three questions, before and after the test, we see that 7 participants indicate an experience with the program better than they expected, as shown in the last column of Table 4.6. These results confirm our expectations about DOGeye design: it is rather easy to use, learn, and it achieves a satisfying fulfillment of COGAIN Guideline 4.2 about graceful and intelligible interfaces.

During the debriefing interviews, we collected some observations from the participants, about their behavior during the study and about what works in DOGeye.

All the participants observed that the name "Home Management" for indicating the tab with the home basic devices is not clear; for example, some of them intuitively looked for lamps in "Electric Device". They suggest to divide the "Home Management" tab in two different tabs: "*Doors and Windows*" and "*Lighting*".

The *tab isolation* feature is "strange" for 7 out of 8 participants: i.e., they expected that once entered in a room in a defined tab, the application "remains" in that room when they change tab. So, they thought tabs as *different views* of the same house, instead of different "virtual houses" with different set of devices in it.

Only 3 users had difficulties to find the general "Alarm" button, placed in the bottom right corner of DOGeye: they look for it in the "Security" tab.

User	Question 1	Question 2	Question 3	Question 4	Better than expected?
E1	4	4	4	4	Yes
E2	4	4	4	5	Yes
E3	4	2	3	4	No
E4	4	4	4	4	Yes
NE1	4	4	4	4	Yes
NE2	5	4	4	4	Yes
NE3	4	5	4	4	Yes
NE4	4	5	5	5	Yes
Summary					
	7 agree 1 strongly agree	5 agree 2 strongly agree 1 disagree	6 agree 1 strongly agree 1 not agree or disagree	6 agree 2 strongly agree	7 yes 1 no

Table 4.6. Qualitative evaluation graded from Strongly disagree (1) to Strongly agree (5).

During the DOGeye study we have deactivated the “Scenarios” and “Everything else” tabs but we have asked participants what they expected to find in them. None understood what “Scenarios” tab includes, thinking about rules, home external views, external lights or music. Three of them, instead, understood the “Everything else” tab and other two subjects found it “useless”.

An interesting thing we noticed is that none of the participants used the “Activate/Deactivate by type. . . ” buttons, thus making the *implicit multiple selection*, present in the first two tabs, unnecessary. A good hint from a participant was to make the “Notification area” interactive, i.e., by offering the possibility to “click” on a notification to implicitly perform a single device selection, allowing the user act on the device.

Comments from users are, in general, very good: some of them appreciate the vocal feedback and the tab divisions, others the presence of the camera in the “Security” tab while some of them just found DOGeye “*very cool*”.

4.8.3 Discussion

Overall, DOGeye evaluation is positive and provides useful insights on home control applications explicitly designed for eye tracking support (COGAIN). By analyzing the success rate of each task, we noticed that tasks like *task #5* ("Turn on all the lamps in the lobby") are the most difficult for users, since they require quite advanced selection modalities. Referring to task #5, participants tend to turn the lamps on one by one, instead of using some kind of multiple selection.

The tab subdivision needs a refactoring as pointed out by user observations: we plan to split "Home Management" in two different tabs: "*Lighting System*" and "*Doors and Windows*". The tab "Everything else" will be removed, since few participants understand its meaning, and the *tab isolation* feature will be removed, thus offering different views of the same house when a user changes tab.

Since nobody used the *implicit multiple selection* present in the first two tabs, that feature will be removed: we keep only the other selection modalities, i.e., implicit selection, single selection and multiple selection. Implicit multiple selection will obviously remain in the "Temperature" tab, since it is the only viable modality for multiple selection in such a tab.

We are continuing to refine DOGeye, by adding functionalities such as a complete implementation of scenarios, a better visualization of the house map or the support for house on multiple floors. Based on our current results, we intend to implement these design and development changes and to conduct a sounder and deeper evaluation in a real smart home setting. We are working towards increasing the amount and quality of interaction for home inhabitants with or without mobility impairments.

4.9 Conclusions

I have introduced *DOGeye*, a multimodal eye-based application that enables people with motor disabilities to control and manage their homes, thus living as autonomously as possible.

The chapter described the basic principle of eye tracking and presented the design and the implementation of DOGeye. I have discussed the various design issues, such as the use of both the structural view and the functional view of a home, by also referring to COGAIN Guidelines for gaze-based environmental control applications. A first user test, with 8 subjects, has been conducted and the relative strengths and weaknesses of DOGeye has been identified, roughly estimating the ability to use the interface without external hints and understanding whether its advanced functionalities are easy to use. Results show that DOGeye can be successfully used through eye interaction and demonstrates only minor weaknesses in its design.

Chapter 5

WristHome: a Wearable Home Access Point

This chapter addresses the problem of unobtrusive user-home interaction by presenting WristHome, a wearable platform for interactions with SmE. Basing upon the viability of wrist watches as interaction means, and applying a strong application-oriented approach, requirements for transforming them into flexible home access points are discussed and formalized. These requirements drive the design, development and preliminary test of an end-to-end system based on off-the-shelf and open source components, named WristHome.

5.1 Motivation

Smart Environments, or Smart Homes, are defined as "*digital environments that proactively, but sensibly, support people in their daily lives*" [7]. Such "sensible and proactive" support is achieved by continuously and unobtrusively complementing human activities. While research-level solutions already support this kind of interaction, although on specific, customized settings, only, few residential homes employ SmE on an everyday basis. Several factors can be identified, which can cause this lack of adoption: (a) the highly-customized nature of research-level solutions, (b) the high installation costs, (c) the inability of existing solutions to actually blend with the inhabitants life background. Human-home interfaces are still under investigation and a suitable tradeoff between traditional (e.g., switches, etc.) and PC/mobile-based interfaces, has not yet been found. Traditional interfaces are well understood, easy to use and not intrusive at all, as they are already part of householder's daily activities. PC/mobile-based interfaces, instead, are typically intrusive and impose additional cognitive load (explicit mediators) on the home inhabitants [72]. Moreover, computers and mobile devices have some known limitations: they are multipurpose devices, and they could be controlled by other home inhabitants (e.g., for gaming); they are not always carried around in the home [19]; they need to be picked up, opened or turned on before they can be used; and, finally, there are situations where is not possible, secure or suitable to use them (e.g., with wet hands or under the shower). As a consequence, their adoption is often confined to small niches, whereas more effective interaction would unveil the full SmE potential.

Wearable computing aims at overcoming part of these user-home interaction issues [73, 74] by enhancing the invisibility of smart systems (e.g., interfaces) and by improving the level of acceptance of proposed solutions in accomplishing home tasks. Wrist watches, or bracelets [75], are among the most attractive solutions for AmI wearable interfaces as they offer a suitable form factor, they have the advantage of always being with users and can be instantly viewed/operated by flicking the wrist. User studies [19] confirm their viability since:

(a) a large fraction of population is already accustomed to wearing watches and/or bracelets;

(b) watches are less likely to be misplaced with respect to phones, tablets or other mobile devices;

(c) watches are more accessible than other devices one may carry;

(d) the wrist is ideally located for body sensors [76] and wearable displays [77].

Unfortunately, also in this domain, actual exploitation is still confined to niches: on one hand research-level solutions are not mature enough to support everyday use, lacking optimization, packaging and wide diffusion, as in [76]; on the other hand commercial solutions are more focused on technologically advanced gadgets, or mobile phone extensions, rather than on fully integrated solutions.

5.2 Requirements

The use of wrist-worn interfaces, such as a watch or bracelet, enables many user-required features for smart homes [8], but poses additional requirements on both hardware and software functionalities. By analyzing the current literature on wearable and pervasive computing, a base set of requirements can be identified (see Table 5.1 for the full list). Watch/bracelet used in smart homes must carry standard sensors on-board, in particular temperature sensors and accelerometers, to exploit user movements [78] and environment conditions (context) in interaction design [79].

Requirement	Description	Priority
Sensing on board	The watch must carry on-board temperature sensors, accelerometers, and, optionally, blood-pressure, and heart-beat sensors	Required
Localization	The watch should support user localization through: RFID, NFC, RF power localization, etc.	Optional
Communication	The watch must provide wireless communication to the home (standard technologies are preferred)	Required
Battery life	Battery must last at least several days	Required
Visual feedback	The watch display must successfully convey information to users	Required
	Multiline display	Required
	Matrix display	Optional
Non-visual feedback	The watch must provide non-visual feedbacks to get the user attention	Required
	Sound emitter	Required
	Haptics	Optional
Touch access	The watch must provide touch-based interaction	Required
	Buttons, touch-sensitive display or bracelet	Required (at least one)
Customization	Aspect customization (color, cover, etc.)	Optional, but typically wanted
	Function customization	Optional

Table 5.1. Wrist-worn User-Home Interface Requirements

Additional features might include blood pressure and heart-beat sensors, which enable home-care and assistive scenarios [76, 80]. By always being on the inhabitants' wrists, watches and bracelets are ideal means for user localization. To accomplish such a task, however, they shall integrate localization technologies such as RFID, NFC, Bluetooth LE, etc. Moreover, they must provide wireless communication to the home while, at the same time, ensuring good battery life, comparable to normal watches [81]. Readability of the watch display and accessibility of the watch buttons (or touch display) is another factor to account, as typical usage scenarios require easy and quick operation [74]. Finally, packaging and software customization enhances the user-home experience, allowing inhabitants to tailor the wrist-worn interface to their specific needs.

5.3 Architecture

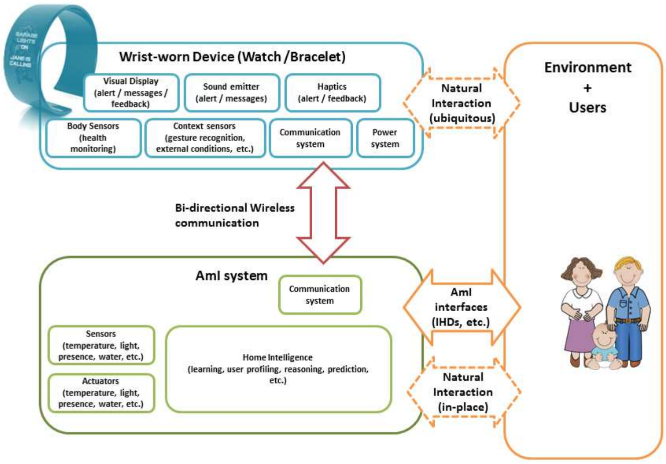

GARAGE LIGHTS ON
JANE IS CALLING
Wrist-worn Device (Watch /Bracelet)
Visual Display (alert / messages / feedback)
Sound emitter (alert / messages)
Haptics (alert / feedback)
Body Sensors (health monitoring)
Context sensors (gesture recognition, external conditions, etc.)
Communication system
Power system
Natural Interaction (ubiquitous)
Environment + Users
Bi-directional Wireless communication
AmI system
Communication system
Sensors (temperature, light, presence, water, etc.)
Actuators (temperature, light, presence, water, etc.)
Home Intelligence (learning, user profiling, reasoning, prediction, etc.)
AmI interfaces (IHDs, etc.)
Natural Interaction (in-place)

Figure 5.1. Logical architecture

The proposed architecture for wrist-worn interfaces (see Figure 5.1) involves three main tiers respectively corresponding to: the wrist-worn device, the SmE system and the home environment, including people living in it. The SmE system and the wrist-worn device tiers are further organized in modules providing functionalities to support user-home interaction, according to requirements identified in the previous section.

5.3.1 Wrist-worn Device

The wrist-worn device depicted in the WristHome architecture is a personal wearable notification, sensing and control device with a bracelet-like form factor. To fulfill the requirements reported in Table 5.1, it exploits a modular hardware and software structure encompassing the following functional modules.

Body sensors (optional): typically encompass blood pressure, heart-beat, and skin temperature sensors. A specific firmware takes care of sampling the corresponding measures and conveying them to the SmE system via the communication system. Some information can also be used locally.

Context sensors (required): such as accelerometers and temperature sensors. They are typically exploited for direct interaction between users and the environment. Differently from body sensors, data flows originated by these sensors are typically processed on the SmE side as the computing capabilities of wrist-worn devices are usually restricted.

Display (required): the wrist-worn device includes a display to provide feedback and information to the user. To fulfill the unobtrusiveness requirements, the display must fade in the background (e.g., shall behave as a normal watch display) and at the same time must be capable to show concise, yet useful, information about the smart environment and the surrounding context.

Sound emitter (required): usually implemented as bi-tonal buzzer. The sound emitter supports immediate feedback and acts as trigger for user attention: whenever a user action must be timely taken to properly face a given home condition (someone at the door, a child asking for help, etc.), the sound emitter drives the user attention to the display, where further information is shown and possible actions are described.

Haptics (optional): the haptic module (e.g., vibration) has almost the same role of the sound emitter but it enables perceivable feedback in all situations were sound is not appropriate, e.g., during meetings, when hearing impairments prevent full exploitation of sounds, etc.

Communication system (required): ensures wireless communication with the AmI environment either through standard (e.g., Bluetooth) or dedicated (e.g., SimpliciTI) protocols. It supports bi-directional communication and optional message prioritization (needed for better handling of alerts).

5.3.2 SmE System

From a very high-level standpoint, 4 main subsystems can be identified: (a) **sensors**, with which the smart home system observes the current environment state and context; (b) **actuators**, used by the system to trigger changes in the environment state, possibly involving the user, both in the decision process and for what concerns the actuation results; (c) the **Home Intelligence**, providing context-awareness, activity recognition, environment operation, proactive interaction, event generation and delivery, etc.; and (d) the **communication system**, able to handle messaging between the SmE and the wrist-worn devices used as human-home interfaces.

5.4 Implementation

The reference architecture has been implemented on a real-world watch, based on the eZ430-Chronos development platform and on Dog [12] (see Chapter 3 for further details), the latter is used to manage the SmE environment.

5.4.1 eZ430 Overview

The eZ430-Chronos is an affordable and complete development system, featuring a 96 segment LCD display and providing an integrated pressure sensor, a 3-axis accelerometer for motion sensitive control, a temperature and a battery voltage sensor. It comes bundled with a USB-based wireless interface which permits to support PC-to-watch communication. Available functions can be reached through 2 menus located on the top and on the bottom row of the watch display, respectively. A standard button operation paradigm is defined for the entire platform, with 3 main interactions: (a) a short pressing of the "#" button switches to the next menu entry, (b) a long (2s) pressing of the "#" button provides access to submenus and finally, (c) a pressure of the "▼" button activates the current menu entry. From the hardware standpoint, the platform fulfills mandatory requirements for wrist-worn interfaces, i.e., availability of sensors on board and capability to wirelessly communicate with a PC-like device (SmE system). On the other hand, the standard firmware provided with the development framework is focused on standard watch functionalities, therefore user-home interaction modules must be designed and integrated as firmware extensions.

5.4.2 Wrist-worn Device Implementation

Watch-level implementation mainly involves the design and development of a firmware extension, starting from the OpenChronos open source version[1]. Visual and sound emitter modules are implemented as new watch functionalities whereas the haptic display module is omitted since the platform does not provide such type of feedback. New functionalities, i.e., gesture recognition (under development), message handling, battery measurement and quick access commands, are included in the watch menu located on the bottom row of the display. Such a menu is, in the eZ430-Chronos firmware design, typically reserved to advanced (non-watch) capabilities such as heart-beat monitoring, mouse control, etc. Interaction between the watch and the smart environment adopts a client-server paradigm and, due to battery saving concerns, takes place either on a sporadic basis (every 30, 60 or 180 seconds) or manually, when triggered by the user. User-home interaction through the watch exploits 3 types of messages:

Silent messages represent low priority messages meant to inform the user about the home state. The watch firmware handles a maximum queue of 2 messages, which are kept in memory until they are overwritten by more recent messages. Received silent messages can be displayed at any time by pressing the "▼" button in the "Message" menu.

Loud messages have the same priority of silent messages, but they solicit immediate user attention by activating the integrated alarm (Figure 5.2). Loud messages share the same memory queue of silent ones, and they are typically used to deliver

[1] `http://github.com/poelzi/OpenChronos`, last visited on January 2014

more urgent information about the home state/context, e.g., anti-burglar detection, help requests, reminders.

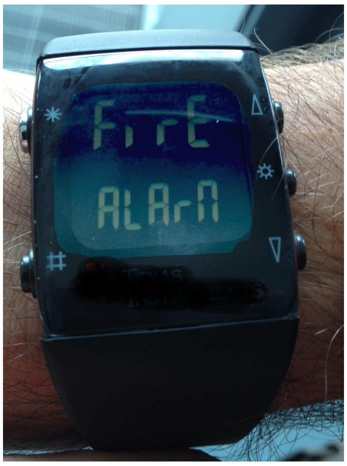


Figure 5.2. A loud message

Reply messages encompass all messages for which a user reply is required, typically in form of a YES/NO answer. They are the highest priority messages exchanged by the watch and the AmI system, and they always require user attention by activating the watch alarm. Possible replies are YES ("▲" button), to activate the suggested action or NO ("▼" button) to avoid it. Any other button pressure is interpreted as don't care.

During normal operation, the SmE system uses the communication module to monitor connections coming from watches distributed in the home environment. Whenever a watch (previously registered with the SmE service) wakes up, the system inspects/updates the message queue for the watch to deliver the 2 most recent messages, giving higher priority to reply messages. It must be noticed that this sporadic operation pattern is typical in battery powered systems, where a suitable trade-off between consumption and responsiveness must be identified.

5.4.3 Wrist-worn User Interface

The display module implemented in the wrist-watch firmware offers 3 master screens with which users interact for accomplishing all message handling tasks. They respectively involve: a main screen, a settings screen and a reply screen. The settings screen is further divided in 2 pages, needed to display all available options on the small LCD screen of the watch.

The **main screen** is the entry point for handling AmI messages. It is identified by the "MESS" string reported on the bottom display line, and can be accessed by iteratively pushing the “#” button. The “▼” button, in this menu, allows manual display of last-received messages, without triggering a watch-to-AmI communication, and thus avoiding quick battery draining due to active wireless link. On the other hand, by holding the “#” button for more than 2s, users can access the message settings pages.

The **settings screen** is organized in 2 pages. The first page is mainly focused on message handling. In particular, it supports watch users in removing the last messages from the message queue, and permits to manually check for any pending messages, without waiting the auto-synchronization trigger. Manual synchronization is the only mean to get AmI messages when the watch is in the manual synchronization mode. In such a case, in fact, no communication is carried unless triggered by the user through the “▼” button, when the watch shows the message settings page. The second page is more focused on watch configuration. On one hand, it permits to enable/disable the sound emitter module, thus avoiding obtrusiveness in all cases where a loud sound might be annoying or inappropriate. On the other hand, it allows selecting the desired auto-synchronization interval, offering three different refresh rates 30s, 60s, 180s and the manual synchronization option.

Whenever a reply message is delivered by the SmE system to the watch, the **reply screen** displays the message while the watch plays a loud sound (if sound has not been disabled), blocking until the user selects one reply option. The user can either choose to ignore the message, by pressing one of the left-side watch buttons, or can explicitly answer: YES, by pressing the “▲” button, or NO, by pressing the “▼” button.

5.4.4 Integration in Dog

The eZ430-Chronos has been integrated in Dog by modeling the relative capabilities in DogOnt (Figure 5.3), and by writing a set of Drivers in Dog.

The set of Drivers implemented in Dog let the gateway communicate with the watch through an extension of the SimpliciTI protocol, one of the protocols natively supported by the wrist watch.

5.5 Experimental Results

A preliminary user study has been carried to evaluate the watch functions and the possible adoption scenarios. Four participants used the system, performing three tasks and replying to a final questionnaire. Their observations and answers were used to carry a qualitative analysis, to help identifying strengths and weaknesses of the system and to identify future directions. The four participants recruited for this

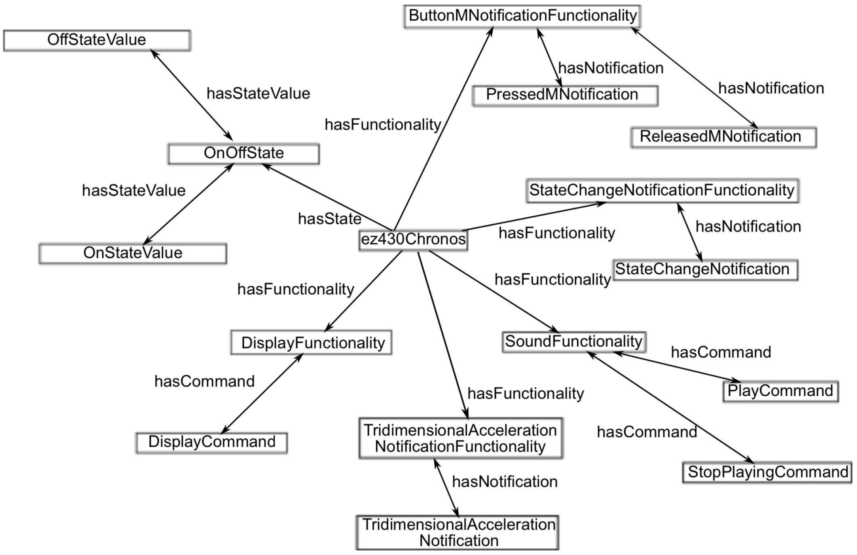


Figure 5.3. The eZ430-Chronos as a DogOnt device

preliminary study were 2 females and 2 males (aged 35-46), of which only one was working in the computer science field. All of them habitually wear a wrist watch.

5.5.1 Environment Setup

The user evaluation has been carried in a controlled environment where Dog acts as the SmE system. In the test environment, Dog controls and receives notification from two different home automation plants, equipped with 6 lamps, 4 mains power outlets, a shutter actuator, and some switches. During the evaluation, Dog sent to the watch two different messages: a request to turn off a lamp and a warning message. Users were required to naturally react to the messages, using the think-aloud protocol for describing their decisions. The watch was able to correctly deliver the messages, forward a reply (when needed) and interacted with the smart environment with no detectable problems.

5.5.2 Qualitative Results

The final questionnaire asked participants to give an overall grade to the system, in a scale from 1 (the worst) up to 5 (the best). Results show that they were quite satisfied of the system behavior and functions, with a mean value of 3.5.

The participants would use such a system in their homes but also in the workplace; moreover, they found the watch menus easy to navigate and to use, but only after an initial explanation. Two of them were quite interested in controlling their home appliances with the watch, whereas the other two participants were very interested in such possibility. All the participants were very interested in the possibility to control their appliances by using some gestures. When we asked for how much they would spend for a watch with such features, three participants said they want to spend 25-50$; the other participant said “less than 25$”. These choices give an indication to an important “cost requirement”, i.e., the wrist watch should be a low cost device, and the adopted development kit might work as a good starting point, by only costing around 50$.

5.6 Conclusions

This chapter discussed requirements for wrist-worn human-home interfaces and proposed a preliminary implementation based on a cost-effective watch. Preliminary user tests confirm the functionality of the system and the viability of the approach. Interesting aspects emerging from user testing involve both the device price, which must be in the low range (between 25 and 50$), and the willingness to adopt the watch in the home and in the workplace. This last observation supports the unobtrusiveness of the approach and fosters future investigations about the possible uses of such an interface.

Chapter 6

RulesBook: Rule-based Activity Delegation

This chapter introduces RulesBook, an easy to use mobile tool that allows users to create rules and context-aware applications to facilitate the delegation of task from humans to their homes, without the need of programming experience. In this way, users can maintain the desired autonomy in their homes, and perceive the SmE as useful and cooperative. The chapter reports the requirements for an effective task delegations, the design and validation of RulesBook, its implementation and a preliminary user evaluation.

6.1 Motivation

Many intriguing scenarios are currently sketching the home of the future, where human inhabitants will only carry out "exciting" or "interesting" tasks and the home will take care of all boring duties that fill our every day life. Future homes will be able to learn our habits, anticipate our needs, supporting, guiding and educating ourselves to a more effective and more environmental-aware interaction with the surrounding world. Although appealing, this long-term vision (part of the SmE research field) has also a worrying connotation where the homes not only facilitate our life but directly modify our home-related behavior in a direction difficult to discern, on the user side. This scenario, already emerging from several studies about user attitudes towards smart homes [82, 83], has been driving an initial research effort on finding suitable trade-offs between totally direct user control and fully automatic home behaviors, involving several degrees of home autonomy, from completely passive solutions to moderately pro-active homes.

No sound and widely agreed solution to this trade-off has currently been found and the related research activities, both in the HCI and SmE communities are still very active. Nevertheless, a relatively accepted approach based on *activity delegation* is gaining momentum. To avoid humans feeling trapped in their own homes, researchers are currently proposing interfaces and tools that allow home inhabitants to explicitly delegate specific, often boring, tasks to the smart environment, requiring the home to autonomously carry out them without further human intervention. Actually, humans are already delegating tasks to their "dumb" homes, for example they delegate thermostats to keep their house temperature comfortable, alarm systems to keep their home safe, and so on. This delegation behavior is already accepted as part of the normal daily routine.

In the SmE community, explicit delegation of tasks to homes is usually realized as rule-definition or user-initiated learning. In the former case, the user is supposed to design/edit one or more home automation rules (also called policies, scenarios) [84, 85]. In the second case, instead, the user puts the home in "learning mode" and teaches the home what behavior must be replayed when a given event or combination of events happens (e.g., by using case-based reasoning techniques [86]). While technology, especially for rule-based delegation, is rather mature and widely investigated, there is still a sensible lack of effective user interfaces. To support users in shaping their specific home automation policies, interfaces must be simple, easy to use and to learn for people without advanced programming skills, and should not require any specific notion about the automation technologies installed in the home. Independently from the smart home solution, be it based on wired or wireless components, integrated by design or retrofitted on existing plants, users must be able to define automatic behaviors simply, on the basis of device states, events and context information, e.g., time, outside weather conditions, etc.

This chapter proposes a tool for overcoming the current lack of effective rule definition interfaces by defining a mobile rule design interface specifically aimed at non-skilled home inhabitants, named RulesBook. RulesBook exploits well established interaction paradigms such as drag 'n' drop, auto-completion and automatic suggestion to provide an easy to use and easy to learn user interface for writing executable rules.

Differently from many approaches where rule components are directly mapped to specific services, RulesBook exploits a constrained grammar to enable users to easily define and edit rules which can directly be injected into intelligent home gateways, equipped with proper rule-execution environments, i.e., rule engines such as JBoss Drools, Jess, etc. The adoption of such a grammar allows, on one hand, to easily "parse" and "convert" user-defined rules in home-executable policies, and, on the other hand, it provides a simple, yet expressive, syntax that can be easily rendered as nearly natural language, thus enabling non-skilled user to better understand the implications of designed policies. The main contributions of this chapter include:

(a) the definition of requirements for user-accessible (i.e., user-friendly, easy to learn) rule editing interfaces in the smart home context;

(b) the definition, design and implementation of a mobile editing interface, enabling ubiquitous access to home personalization and programming functionalities;

(c) the design of a simple-to-use, yet expressive visual language for rule creation specifically targeted at non-expert users, empowered by a formal grammar.

Such novelties are complemented by a careful application of HCI design principles which is reflected in the presented design and preliminary user validation, that confirm the effectiveness of the approach.

6.2 Requirements

Delegating part of everyday tasks to the home requires suitable interfaces for enabling the home inhabitants to easily define processes to be automated, i.e., to effectively program automation rules. Typical computer-based rule languages (e.g., JBoss Drools, Jess, SWRL) have great flexibility and expressiveness but require profound knowledge about the context in which rules are implemented and about the syntax and composition grammar associated to the specific language. Clearly, this high level of required skills prevents typical home inhabitants from directly editing such rules, thus limiting their ability to wittingly delegate tasks to their homes. By interacting with both people living and managing smart homes[1] and with people commercializing wired and wireless home automation systems[2], we derived the following set of requirements that an effective rule builder interface shall obey.

1. Rules shall be **definable by people with basic level of computer literacy**, the only required knowledge is about the home components, in terms of normal usage and behavior.

   (a) Home *devices shall be exposed in an abstract and technology independent way*, thus enabling user to easily specify the rule objects.

   (b) *Rules shall be self-explaining*, i.e., they can be directly/easily translated in a nearly natural language description, e.g., by providing a readable rule summary.

[1] the Maison Equipée rehabilitation structure in Valle d'Aosta, and the C.E.T.A.D. center for assistive technologies of the Turin municipality

[2] we are KNX Scientific Partners, and we collaborate with BTicino (MyOpen) and other home automation technology vendors

   (c) *Rules shall always be "valid"*, i.e., the user can only create and save syntactically (and possibly semantically) correct rules.

   (d) Rules shall be *expressive enough to manage most situations, actions and interactions* that a home inhabitant may want to delegate (i.e., they shall be easily mapped onto a powerful enough computer-based rule language).

2. Rules shall be defined in various places of the environment and, possibly, in **mobility**.

3. The **rule-design interface must facilitate** the **delegation of tasks** from humans to homes **providing suitable "aids"**.

   (a) Rules editing shall be facilitated by means of *suggestions, guiding interfaces* and *auto-filling* functionalities.

   (b) Rule interface should offer *support to handle unexpected loss of connections* or *computer malfunctions*, e.g., *automatically saving rules.*

6.3 Design

This section proposes the RulesBook interface concept fulfilling the previously described requirements. The interface mockup has been initially design for the Web and, afterwards, ported to a native Android app. For the sake of clarity the section first reports the interface design concept and then it focuses on the formal grammar that lies at the basis of the approach. Eventually, the section present the final version of the RulesBook interface.

6.3.1 Concept and Use Case

Sam, a smart home user, takes its tablet and open RulesBook. He wants to create a rule to turn on the lamp in the living room when the room is dark. The interface he sees on his tablet is sketched in Figure 6.1. On the left, he sees the two devices he wants to use for creating the rule (a lamp and a light sensor) and other two objects: a *clock* and an icon labeled "*Everything by type...*". On the right, he can see a wide area area to be used for the definition of a rule. The dotted rectangles under the IF and THEN keywords are strong visual clues suggesting to drag a device inside them (req. 3a).

Sam tries to imagine a new rule as a sentence that starts with IF and that uses THEN, like "*if the light intensity is low, then turn on the living room's lamp*". So, he drags the light intensity sensor under the "IF". When the sensor icon is over the dotted rectangle, it docks under the "IF" as a rectangular container. In this

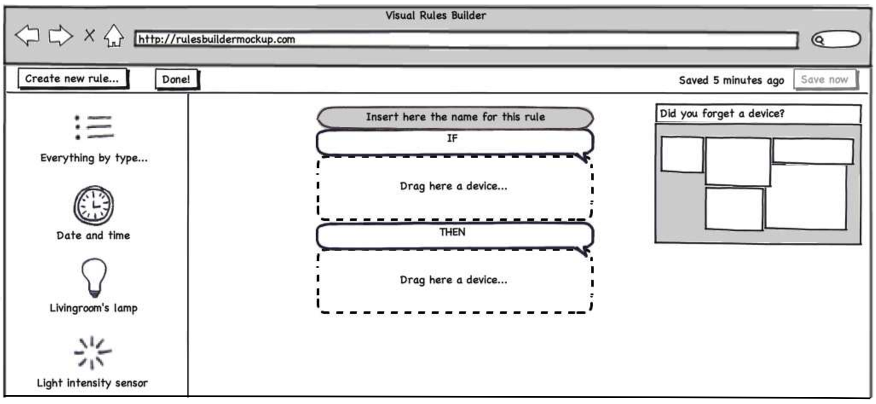


Figure 6.1. The RulesBook Start Page

container, besides the sensor name, Sam also sees a list to specify what sensor event should be intercepted. Sam chooses "LOW" light intensity (req. 1a). By looking at the right area of the interface, Sam notices that something has changed: two other rectangles appeared before the "THEN" keyword, as shown in Figure 6.2 (req. 3a and 1c).

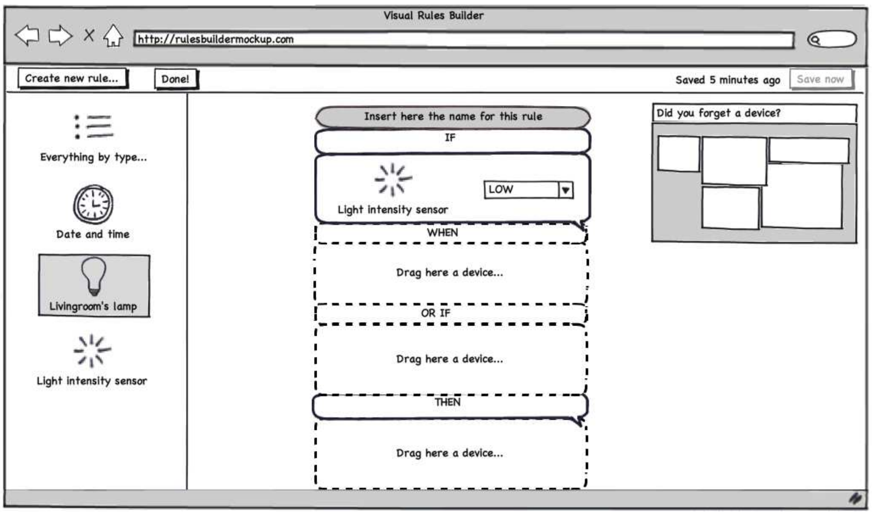
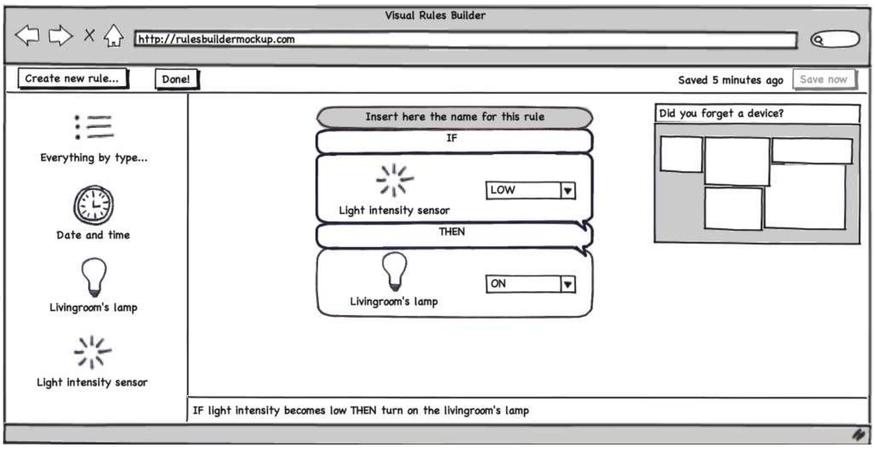


Figure 6.2. What happens when the "IF" area is complete

The two new dotted rectangles are the optional "WHEN" and "OR IF" statements (req. 1b and 1d). Sam understands their meaning but does not need them in his rule. He decides to drag the lamp icon under the "THEN" keyword. Even in this case, the icon docks on the dotted rectangle as a rectangular container. Sam, as before, selects "ON" between the options presented by the lamp container (req. 1a). The rule is complete and the lower part of the interface reports a sentence that summarizes the just created rule (Figure 6.3): *IF light intensity becomes low, THEN turn on livingroom's lamp* (req. 1b).



Light intensity sensor: LOW

Livingroom's lamp: ON

IF light intensity becomes low THEN turn on the livingroom's lamp

Figure 6.3. Complete rule

Sam, now, wants to save the newly created rule. By looking for a "save" button, he finds it in the upper right corner of the interface but he notices that is not active and a label informs him that the rule is already saved (req. 3b). Sam assumes that the rule has been auto-saved during its composition. In the end, he adds a rule name, and closes the rule by clicking on the "Done" button.

6.3.2 Grammar

The RulesBook concept just illustrated guarantees rule correctness (req. 1c) and readability (reqs. 1b, 1d and 3a) by exploiting a formal rule representation grammar (see Figure 6.4) based upon four fixed keywords: IF, THEN, WHEN, OR IF (req. 1c and 1d). The first two are mandatory for the creation of any rule, while the others are optional (dotted in Figure 6.4). A rule composed with this grammar follows the natural language (req. 1b). An example of such rule could be: "IF a window of the kitchen has been open WHEN the heating system is on, THEN turn the kitchen's heating system off".

The IF keyword expresses an event to trigger the rule. The event is indicated, in Figure 6.4, as an "E-BLOCK" (event-block). WHEN defines one or more conditions constraining the event; multiple constraints should be simultaneously satisfied. The set of constraints is shown as "C-BLOCKS" (constraint-blocks). OR IF is a disjunction for repeating the IF-WHEN part more than once. Finally, THEN indicates a set of actions to be executed on the occurrence of the above triggers. The actions are indicated as "A-BLOCKS" (action-blocks).

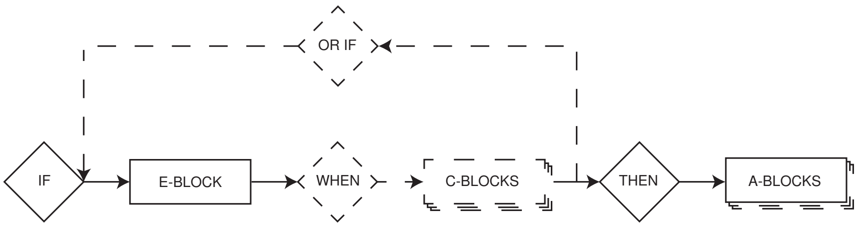


Figure 6.4. The grammar underlying the creation of a rule

To maintain rule consistency (req. 1c), each device involved in the creation of a rule has a different behavior according to the block in which is inserted, as described below. For example, the A-BLOCK does not accept non-controllable device, such as sensor.

E-BLOCK interprets events generated by controllable devices, clock and sensors. Controllable devices are shown as "*device-name* becomes *state*"; the clock expresses a temporal event and is shown as "time is equal to *HH:MM* (on *Weekday*s)". Sensors are shown as "*sensor-name* becomes lower/higher than *threshold-value*".

C-BLOCK supports controllable devices, clock and sensors. Controllable devices are shown as "*device-name* is *state*"; if it is a clock, to express a temporal interval, it is shown as "time is between *HH:MM* and *HH:MM* (on *Weekday*s)". If the inserted device is a sensor, it is shown as "*sensor-name* is lower/higher than *threshold_value*".

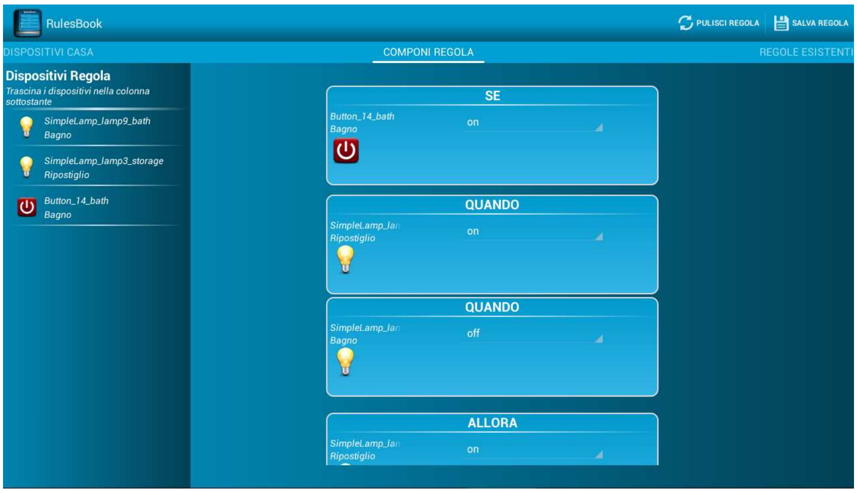
A-BLOCK supports controllable devices only: they are shown as "*command device-name*" or "*command device-name* for a *time-interval*".

The interface concept and the RulesBook grammar have been informally verified, and approved, by a restricted focus group.

6.3.3 The RulesBook Interface

The final appearance of RulesBook is shown in Figure 6.5, it is possible to notice three areas, similar to the interface mockup.

- *Navigation area*, in the upper part of the application. It indicates the current page and shows the previous and next pages. Movement between pages occurs by swiping on the left or on the right, thus moving on the previous or next page.



Dispositivi Regola
Trascina i dispositivi nella colonna sottostante

SimpleLamp_lamp9_bath
Bagno

SimpleLamp_lamp3_storage
Ripostiglio

Button_14_bath
Bagno

SE
Button_14_bath
Bagno
on

QUANDO
SimpleLamp_lar
Ripostiglio
on

QUANDO
SimpleLamp_lar
Bagno
off

ALLORA
SimpleLamp_lar
Ripostiglio
on

Figure 6.5. The RulesBook application

- *Collection area*, a column that may be in the left or in the right part of the interface. It stores a list of objects needed in the current page. In Figure 6.5, it is placed on the left and shows the list of the devices selected for composing a rule.

- *Action area*, a wider area that may be in the right or left part of the interface, according to the position of the previous area. It let users perform some types of actions or selections. In Figure 6.5, this area is placed on the right and shows the composition area of a rule.

Three pages are available in the application: a **Home** page for selecting the devices to be used for composing a new rule or editing an existing one; a **Rule Builder** page for effectively compose the rule, with the same mechanisms reported in Subsection 6.3.1; and a **Existing Rules** page for selecting or removing already existing rules.

The application is localized in Italian and encompasses more functionalities than the mockup prototype, such as the management of existing rules or the selection of devices. Differently from the mockup, the current version of RulesBook does not implement the *clock* and the "*Everything by type...*" objects.

6.4 Implementation

RulesBook has been realized as an Android 4.x application for tackling the *mobility* requirement (req. 2).

The application is composed by a unique Activity that comprises a ViewPager for handling the active page content, managing the interaction between the pages and the overall navigation. Each page is represented by a Fragment inside the ViewPager. Such an organization allows developers to easily edit the various page, changing their relative position, or adding new pages to include new features.

The RulesBook app is connected with the Dog gateway through the REST API offered by the gateway itself. Rules composed with this application can be parsed and, then, used by the *Rule Engine* addons briefly described in the Chapter 3. In this way, the application hides the low-level knowledge required to operate a specific rule engine.

RulesBook was tested on an *Asus Transformer Pad*, a 10” tablet running Android 4.1.

6.5 Evaluation

The goal of the initial user evaluation is to have a first feedback on the viability of the proposed approach and, at the same time, to identify the relative strengths and weaknesses of RulesBook as a tool for delegating task to the home.

Six participants used RulesBook in a controlled environment, performing five tasks each. Test tasks have been extrapolated from the requirements and use case reported in this chapter, and they represents common operation that a user need to perform in creating or editing a rule. Participants never met during the evaluation. Their observation were used to allow qualitative analysis, to help identify strong and weak points of the interface, and to help identifying future directions.

Test Deployment A total of six subjects volunteered to participate in this evaluation, with ages ranging from 20 to 35 years, all males. Participants had a variety of educational backgrounds. They were required to have prior general exposure to smartphones or tablets.

After a short introduction to the test and the collection of demographic data, an overview of the RulesBook app has been provided. Each participant, then, was told to complete a set of five tasks, one at time. For all of them, the *think-aloud* protocol has been adopted, to verify their actions. Examples of proposed tasks includes “Take 12 devices and create the following rule [. . .]” and “create a rule that involves two devices”.

At the end, participants were given a questionnaire and asked to rate RulesBook in general and by querying their agreement level (from “Strongly disagree” to “Strongly agree”) about these sentences:

1. I think the application is intuitive.

2. I think that the application layout is understandable.

3. It was easy to learn how to use the application.

Users open comments or explanations were collected (e.g., difficulties found or explanation about something done during a task) through debriefing interviews. The duration of the entire experiment ranged between 10 and 20 minutes.

6.6 Preliminary Results

The final questionnaire asked participants to give an overall grade to RulesBook, in a scale from 1 (the worst) to 5 (the best). Results showed that they were satisfied from the app, with a mean value of 4.

In the test conclusion, users were also asked to express their agreement about three sentences. Results were quite satisfying: all the users agree or strongly agree with the proposed sentences about RulesBook.

Moreover, all the participants completed with success all the five tasks. Only the mechanism for moving between pages (i.e., the horizontal scrolling) has slowed down three users, for the absence of a clear indication on how to perform such an operation. The operation of creating a new rule, main focus of RulesBook, went smooth, without any problems or difficulties.

All the minor bugs and possible improvements, emerged from the debriefing sessions and from the test execution, were already applied to the current version of RulesBook.

6.7 Conclusions

This chapter presented RulesBook, a tool for tackling the current lack of rule definition interfaces for smart home environments. RulesBook specifically targets non-expert users, i.e., home inhabitants with little or no technological skills.

The chapter distilled the basic requirements of home rule development environments, introducing the conceptual design of RulesBook, a interface fulfilling the reported requirements, and its implementation as an Android app. Results from an initial user test seem to confirm the viability of the approach and the intuitiveness of the application.

Chapter 7

Increasing Energy Consumption Awareness: a User Survey

This chapter try to tackle the design of an effective user feedback display on energy consumption by defining two possible visualization paradigms, based upon principles well-known in literature, and by carrying a web-based survey on the widest possible audience to gather insights on strengths and weaknesses of the proposed solutions. In addition, it also gather insights on the preferred location for in-home energy displays (IHDs).

A total amount of 992 users participated in the survey, mostly from Italy with contributions from Spain, Finland and USA. Survey results provide interesting insights about the analyzed IHD visualizations, and prepare the field for the application presented in the next chapter.

7.1 Motivation

In the last years, energy conservation and sustainable living gained ever increasing attention fostered by many factors including the political situation, economic stagnation, greener lifestyles and philosophies. Counterintuitively energy conservation, in developed countries, is currently more related to residential houses than to industry and commercial production. Homes, in fact, are becoming one of the major contributors to the countries energy balances, as demonstrated by statistics provided by the energy departments of USA and European Union. Typical forecasts for home energy consumption show ever increasing and worrying figures, that in the near future will probably exceed 40% of the total yearly consumption [87], in most of the western countries. This increased awareness fosters and motivates many research efforts on saving energy at home, ranging from making homes smarter and more energy friendly to increasing awareness of home inhabitants inducing important behavior changes in the daily routines of households. Householder awareness, in particular, has a saving potential of around 5%-15% [43], meaning that just by slightly changing their daily behaviors, home users can save up to 15% of their current energy needs. However, convincing people to change daily routines is not trivial and can seldom rely on almost static, monthly information written on paper bills or on-line energy accounts. Direct, real-time feedback is needed, instead, to constantly inform users about the energy efficiency of their customary behaviors with the aim of teaching home inhabitants more environmental friendly ways of living.

This chapter reports the results of a web-based survey that has been carried during the initial phase of a IHD design, as part of a wider effort on applying user centered design methodologies to the whole design process of in-home displays, including their interactions with existing home automation systems.

The survey goal is to validate two different visualization and interaction modalities for increasing electric energy-consumption awareness (energy goal setting and direct power feedback) against the needs of a wide user base (992 users) of technology-aware[1] people living in a home. We designed two prototype interfaces respectively implementing direct visualization of currently absorbed electric power and goal-setting for the electric energy consumed in a day. We required users to watch and analyze two simple video mock-ups, and to respond to a set of carefully designed questions (Section 7.4), aiming at:

a) understanding whether people better comprehend and accept energy goal setting or direct power consumption visualizations;

b) verifying/confirming the willingness of surveyed users to actually adopt an in-home energy display;

c) checking if color-based feedback, i.e., feedback using color variations in parallel with explicit numbers[2], is effective in conveying information about energy/power consumption;

d) evaluating room-level repartition of goal and power data, verifying if corresponding visualizations are easy to understand by users, if such information is felt useful and if more (or less) detail is needed;

e) gathering the users' preferred setting and position for IHDs.

[1] we define technology-aware people as persons habitually using basic web technologies (browser and e-mail)

[2] which are required anyway, e.g., for enabling color-blind persons to use the IHD

7.2 Feedback, User Behavior and Saving Strategies

To understand how an in-home energy display (IHD) may affect the home inhabitants habits, promoting positive changes in terms of energy efficiency and environment-friendly behaviors, it is important to frame the typical user behaviors related to energy consumption (or saving) and to understand the interaction paradigms lying at the basis of currently available solutions. The following subsections provide a brief overview of typical home user behaviors, with respect to energy saving, and the possible saving strategies that IHDs can exploit/induce.

7.2.1 Energy Saving Behaviors

Home displays aim at changing householders behavior to be more energy efficient and environmental friendly. Literature studies show that this increased energy efficiency can be achieved by acting on two distinct classes of behaviors: efficiency behaviors and curtailment [88]. Efficiency behaviors are typically performed once, e.g., by substituting an obsolete refrigerator with a new A+ class one, and their effects usually last for long periods of time (permanent or semi-permanent). On the other hand, curtailment refers to repetitive behaviors that householders adopt to reduce their energy consumption, e.g., turning off the personal computer when nobody uses it. Differently from efficiency behaviors, curtailment requires constant efforts by the home users and is typically targeted by most of IHD designs. Although its impact on the overall savings is generally lower than that of efficiency behaviors, it is still important because it does not require changes in the home environment and because it is subject to the rebound effect, which might invalidate saving efforts. The rebound effect occurs when a home inhabitant uses a new appliance much more than the older one, due to its higher efficiency. The end result is no overall change, or worse, an increase in energy usage.

7.2.2 Energy Saving Strategies

Many strategies have been proposed to tackle efficiency and curtailment behaviors, and they can be roughly categorized in 2 main families: *antecedent* and *consequent* strategies. Antecedent strategies are designed to induce or to avoid a user behavior, consequent strategies, instead, are designed to inform the user after the behavior occurred.

In the former category a sufficiently wide consensus [89, 90, 88] has been reached on: *Information*, *Goal setting* and *Commitment*. *Information* strategies provide residents with information and tips on how to reduce current energy consumption, how to select more energy efficient appliances, etc. *Goal setting* strategies exploit the natural competitiveness of humans to stimulate householders to reach a self-imposed (or interface suggested) energy goal, lower than the current energy consumption. *Commitment* strategies ask home inhabitants to explicitly commit to energy conservation measures. Although similar to goal setting, to which is often combined, commitment differs from goal setting on the psychological side: while goal setting pushes the user towards better behaviors, without requiring clear and voluntary acts, commitment requires users to explicitly and "rationally" adhere to energy reduction policies. Among antecedent strategies, goal setting reached a relatively wide consensus showing real potential to induce reductions in absorbed energy, from 2%-5% up to 20% [43].

Consequent strategies typically include three widely agreed approaches: *Feedback*, *Reward* and *Criticism*. *Feedback* shows residents how much energy they use; it can assume different forms and it must be easy to understand and immediate in its effects, i.e., users shall be enabled to immediately relate provided (visual) information with the corresponding home set-up. *Reward* consists in providing users rewards (monetary or social) for their good energy behaviors. Finally, *Criticism* is based on the idea of confronting users with surrounding people, passing judgments on them that depend on how well do they save energy in the home. This last mechanism proved to be rather unstable in its effects with many studies providing contrasting results. On the converse, feedback is widely recognized as a viable solution whilst reward has been relatively less investigated due to the difficulty of convincing energy providers to support monetary incentives for better energy behaviors and to the inability of finding reliable enough immaterial rewards such as reputation.

7.3 Survey Focus

In order to achieve successful results in guiding home users towards achieving sensible energy savings, we concentrate on the two strategies currently attracting more consensus: goal setting for what concerns antecedent strategies and feedback for consequent approaches. We consider the two approaches as complementary elements of the same interface concept[3], with the aim of teaching users how to best perform with respect to efficient energy consumption. While goal setting aims at preventing bad behaviors by imposing a competitive "pressure" on the home inhabitant, feedback aims at supporting the home inhabitant in understanding its current behavior,

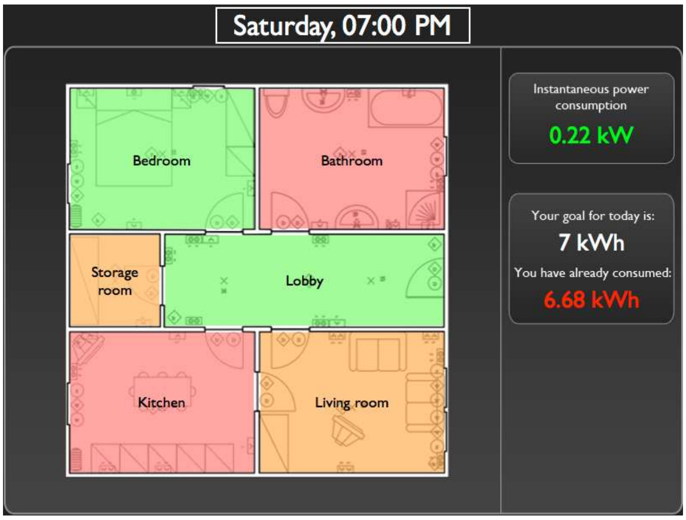
[3] even though more strategies can be combined together, we deliberately choose to adopt only two strategies in order to avoid information overload, which might inhibit positive results as pointed out by Wood and Newborough [42]

in highlighting wrong or not-efficient habits and in taking the needed corrective actions.

In-home displays showing energy consumption require a set of basic assumptions on the home environment in which they are installed: the presence of one, or more, energy or power meters, the possible availability of home automation devices, the display size and placement, etc. The survey presented in this chapter is based on the following assumptions:

- the availability of one meter per room or of an equivalent metering system able to provide measurements at room-level granularity;
- the availability of a home automation plant able to detect and report home device activations;
- the availability of a medium-sized (e.g., 7” or greater) display hardware.

By building on top of this hypothetical but realistic home set-up, we define two different visualizations sharing the same visual layout (shown in Figure 7.1, where only one interface is presented as the layout and visual appeal of both visualizations is very similar) and focused respectively on direct power feedback (DPF) and energy goal setting (EGS).



Saturday, 07:00 PM

Instantaneous power consumption 0.22 kW

Your goal for today is: 7 kWh

You have already consumed: 6.68 kWh

Figure 7.1. The proposed visualization layout for IHDs

Interface features common to both visualizations include:

(a) a clock display showing the current time: this allows users to correctly perceive time and permits to correlate interface changes with the corresponding temporal information;

(b) a colored home map showing the home rooms in color nuances ranging from green (good performance) to red (bad performance), depending on the current power or energy consumption;

(c) a numeric indicator reporting the electric power currently absorbed by the home, colored from green to red as the consumed power approaches the maximum power allowed for the home[4];

(d) a couple of numeric displays showing the energy goal to be reached in a day (or in a week) and the currently consumed energy, also in color hues ranging from green (good) to red (consumed energy approaching or exceeding the current goal).

Room coloring on the home map is dictated by two different algorithms: a direct power feedback strategy (DPF) relating the power currently consumed in a room with the maximum power allowed for the whole home, and a goal setting strategy (EGS) where room color information depends on how currently consumed energy is distributed among rooms, with respect to the energy goal set for the whole home.

In both cases, we first define the fraction $F(r)$ of power consumption allocated to a room r from the set of all rooms R. Such fraction takes into account the set of devices installed in the room (or that may be used within the room), compared to the whole house. In the current experiments, the room fraction $F(r)$ has been computed according to (7.1), where D is the set of all devices d, and is partitioned among devices that can me moved across rooms D_m (e.g., the vacuum cleaner) and devices permanently installed in a room r: $D_f(r)$. $P_D(d)$ is the nominal power of a device $d \in D$ and $|R|$ is the number of rooms in the home.

$$F(r) = \frac{\sum_{d \in D_f(r)} P_D(d) + \frac{\sum_{d \in D_m} P_D(d)}{|R|}}{\sum_{d \in D} P_D(d)} \tag{7.1}$$

In the DPF case, every room is assigned a share of maximum power $P_R(r)$, computed by scaling the maximum allowed power for the home P_M by the room fraction $F(r)$, as in equation (7.2).

$$P_R(r) = P_M \cdot F(r) \tag{7.2}$$

At runtime, every room in the home map changes its color (green, orange, or red) depending on the ratio of its actual current power consumption $P_A(r)$, compared with the room power share $P_R(r)$, according to easy to tune thresholds (7.3).

[4] i.e., the maximum power permitted by the delivery contract

$$\frac{P_A(r)}{P_R(r)} \in \begin{cases} [0,\alpha)\,, & \text{green} \\ [\alpha,\beta)\,, & \text{orange} \\ [\beta,1]\,, & \text{red} \end{cases} \tag{7.3}$$
$$0 < \alpha < \beta < 1$$

In the presented survey α was chosen as equal to 0.4 and β equal to 0.8. Other values may be selected, as well; the survey is, in fact, designed to have a low sensitivity of results with respect to these tunable parameters.

In the EGS case, i.e., the energy goal setting strategy, every room on the home map changes color depending on the amount of energy consumed in the room with respect to the goal quota assigned to the room. Given the overall energy goal E_G assigned to the home over a time period, every room is assigned a goal quota $E_R(r)$ proportional to the room fraction $F(r)$, as in (7.4). Similarly to the power case, the energy consumed by each room, during the goal validity time frame, is compared with the goal quota assigned to the same room and the resulting color hue is computed using the same threshold policy used in the direct power feedback visualization, where P components are substituted by E values.

$$E_R(r) = E_G \cdot F(r) \tag{7.4}$$

7.3.1 Reference scenario

To better illustrate the two strategies reported in the previous subsections, we have built two short videos to be used in our web-based survey. The two videos are based on the same house model. The modeled house is a flat composed of six rooms: a kitchen, a bathroom, a living room, a lobby, a bedroom and a storage room. Rooms contain different devices and appliances, whose power consumption is reported in Table 7.1. Moreover, there is a "mobile" electric device (i.e., the vacuum cleaner) that is considered differently from the other statically installed appliances. We are aware that the environment appliances listed in Table 7.1 might change depending on different cultural contexts. However, in this study, we are mainly interested in evaluating the proposed interaction paradigms, and the user reactions to the provided feedback information. In simpler terms, it does not matter too much what specific consumption users see, but how they perceive it and how they react to the provided information.

Power figures reported in Table 7.1 reflect realistic device consumptions extracted from the "Your Electric Appliances" report, edited by *Seattle City Light*[5]. For

[5] `http://www.seattle.gov/light/conserve`, last visited on January 2014

Rooms	Devices	Consumption
Kitchen	Electric Oven	2200 W
	Microwave Oven	700 W
	Fridge	150 W
	Neon Lamp	11 W
	Dishwasher	1200 W
	Coffee Maker	1000 W
	TV	60 W
Bathroom	Washing Machine	2250 W
	Lamp	15 W
Living Room	Stereo	80 W
	Lamp	15 W
	DVD Reader	20 W
	TV	60 W
Lobby	Lamp	15 W
Bedroom	Ceiling Lamp	80 W
	Alarm Clock	7 W
	Notebook	70 W
Mobile	Vacuum Cleaner	1500 W

Table 7.1. Devices and appliances present in the house model, with their consumptions

devices not present in the Seattle City Light's list, we have acquired nominal wattage from real appliances installed in our homes.

Direct Power Feedback

The video showing the behavior of our house model in the "direct power feedback" case represents a typical day in the life of flat inhabitants, where the maximum power available in the house is 3 kW. The video lasts 1 minute and 50 seconds and covers different activations all day long. We present devices' activations for 5 minutes every six hours (focus points), accelerated 12 times to maintain the video (and the entire questionnaire) as short as possible. In the hours between every focus point, devices keep turning on and off, thus motivating rooms color changes.

As an example, consider the following video fragment: at 12:00 PM, the rooms in the IHD are green and the total instantaneous power used in the house is 160 W; the only active devices are the fridge and the alarm clock. At noon, someone turns on the TV and the microwave oven in the kitchen. At 12:01 PM, the IHD shows the kitchen colored in orange and the total power consumed is 760 W, when the fridge consumes less. This situation persists until 12:05 PM. The entire video storyboard is summarized in Table 7.2, where room names are abbreviated, and G represents the green, O the orange and R the red colors.

Time	**What happens?**	**Room colors**					
		Kit.	*Liv.*	*Bat.*	*St.*	*Bed.*	*Lob.*
06:00 AM	Someone turns on the lamp and the coffee maker in the kitchen	G	G	G	G	G	G
06:01-06:05 AM	The lamp and the coffee maker are active	O	G	G	G	G	G
12:00 PM	Someone turns on the TV and the microwave oven in the kitchen	G	G	G	G	G	G
12:01-12:05 PM	The TV and the microwave oven are active	O	G	G	G	G	G
06:00 PM	Someone turns on the washing machine in the bathroom	G	O	O	G	G	G
06:01-06:04 PM	The washing machine and the other devices previously on are still active	G	O	R	G	O	G
06:05 PM	The washing machine starts to consume less power	G	O	O	G	O	G
The fridge and the alarm clock are always active. The fridge cycles its power consumption every minute.							

Table 7.2. The video storyboard for the instantaneous power visualization

Energy Goal Setting

The video reporting the behavior of the house model in the goal setting representation shows a typical day in the same household of the previous case, where the daily energy goal is set to 7 kWh. The video lasts 1 minute and covers different activations all day long. In particular, we present a "snapshot" of the energy consumption in the house each hour. Moreover, to better appreciate the energy variation occurring in the house, we show the first 2 minutes every six hours (accelerated 12 times, as before). In the other hours some devices turn on/off, inducing changes in the rooms colors. This difference of shown time interval, compared to direct power feedback visualization, is needed for better representing the daily evolution of our house model.

As an example, consider the following scenario: at 07:00 PM, the previously switched on washing machine and notebook are turned off. At the same time, in the kitchen, the lamp and the TV are turned on. The total amount of energy used in the house up to this moment is 6.68 kWh. The bathroom and the living room change color: the former becomes red while the latter is orange. This situation remains the same up to 09:00 PM. On the end of the day, the house inhabitants will exceed the daily goal. The entire video storyboard is summarized in Table 7.3, where rooms names are abbreviate, G represents the green color, O the orange and R the red color.

7.4 Survey design and planning

The definition of the type of feedback and the information to show to users, allowed us to design a web-based questionnaire to collect opinions and needs of home inhabitants. The primary reason for this approach, as opposed to face-to-face or telephone interviews, was that we aimed at reaching as many people as possible while maintaining costs as low as possible, letting users to answer our questions in their preferred times.

The survey was localized both in Italian and in English, and was kept open from September 27, 2010 up to January 31, 2011. It required about 15 minutes for completion. The questionnaire targeted technology-aware participants with a normal domestic life experience. No particular knowledge about energy consumption and measurement were required. For this reason, the survey was announced via emails and social networks (e.g., Facebook, Twitter, LinkedIn, etc.) to colleagues and friends, on the students mailing list of some universities and on the ACM CHI-WEB mailing list. By using these distribution methods, we expected to reach a significant number of people, in Italy and abroad.

7.4.1 Survey form

To encourage high survey participation and completion, we carefully considered the global design of the survey, the formulation of the asked questions and the layout of these questions. Survey replies were anonymous and each respondent could complete the questionnaire once. To reduce misunderstandings due to language barriers we decided to build two versions of the same questionnaire: one in Italian and the other in English. This choice allowed Italian people without fluent knowledge of the English language to successfully understand and complete the survey.

Questions were divided in 4 groups and, for each question in a group, a set of 4 to 5 answers were provided with at least one answer completely wrong and one completely right. Questions involving aspect that we felt critical for the survey

Time	**What happens?**	**Room colors**					
		Kit.	*Liv.*	*Bat.*	*St.*	*Bed.*	*Lob.*
00:00 - 05:00 AM	Everything is off*	G	G	G	G	G	G
06:00 AM	Someone turns on the lamp and the coffee maker in the kitchen	G	G	G	G	G	G
07:00 AM	The coffee maker in the kitchen is turned off	G	G	G	G	G	G
08:00 AM	The lamp in the kitchen is turned off	G	G	G	G	G	G
11:00 AM	Someone turns on the TV and the oven in the kitchen	G	G	G	G	G	G
12:00 PM	The TV is still active and the microwave oven is turned on	O	G	G	G	G	G
01:00 PM	The TV is still active and the coffee maker is turned on for 12 minutes	O	G	G	G	G	G
02:00 PM	The dishwasher is turned on for one hour	R	G	G	G	G	G
03:00 PM	The vacuum cleaner is turned on for 5 minutes in each room	R	G	G	G	G	G
04:00 PM	The TV in the living room is turned on	R	G	G	O	G	G
05:00 PM	The TV in the living room is still on and the notebook is turned on	R	G	G	O	G	G
06:00 PM	The TV in the living room is turned off, the notebook is still on and the washing machine is turned on	R	G	G	O	G	G
07:00 PM	The TV and the lamp in the kitchen are turned on	R	O	R	O	G	G
08:00 PM	The TV and the lamp in the kitchen are still on for this hour	R	O	R	O	G	G
09:00 PM	The dishwasher in the kitchen the TV and the lamp in the living room, the lamp in the bathroom are turned on	R	O	R	O	G	G
10:00 PM	The TV and the lamp in the living room are still on, and the lamp in the bedroom is turned on	R	O	R	O	G	G
11:00 PM	Everything is turned off*	R	O	R	O	O	G
*The fridge and the alarm clock are always active.							

Table 7.3. The video storyboard for the goal-based (energy consumption) visualization

success were usually duplicated in different forms, to cross-check answers, and suggested responses allowed for a certain degree of flexibility in the answering process, supporting partially right or partially wrong statements.

Our questionnaire was composed of an introductory description followed by the four question groups:

1. *Warm up…*, to collect some personal information;
2. *Direct power feedback*, to collect information about the IHD showing DPF information;
3. *Energy Goal setting*, to collect information about the IHD showing EGS data;
4. *Final rush…*, to collect users' preferences and suggestions.

The next subsections detail the different group contents. Questions reported in the following tables and marked with "M" are *mandatory*; the ones marked with "O" are *optional*, and the ones marked with "A" are *alternative* to each other, i.e., they are randomly shown to different users.

Warm up…

In this question group, we gathered some demographic information, such as age, job and country where users live (Table 7.4). The answers in this group are free text, except for the gender.

Warm up…	
1. How old are you?	M
2. Gender?	M
3. What is your job?	M
4. Where do you live? Please, write the country.	M

Table 7.4.   The questions proposed in the "Warm up" group

Direct power feedback

For this group, users were asked to first watch the video showing DPF information about our house model. After the video, they have to reply to six questions (Table 7.5), all with multiple choice answers. Questions marked as "alternative" are presented in a random order, two at a time.

We expected participants to be able to understand all the implicit and explicit activations of the devices in the house model, by carefully watching the video. Moreover, users should be able to estimate the maximum power allocation defined for the house and understand how the consumption changes. Questions 4-6 referred to people's understanding of room colors. We suppose that most participants are able to comprehend why a room becomes green, orange or red.

Direct power feedback	
1. What could be the maximum power allocation defined for the home in the video?	M
2. When was the power consumption highest?	M
3. What appliance consumed most power?	M
4/5. A room is green if...	A
4/5. A room is red if...	A
4/5. A room is orange if...	A
6. Do the red, orange and green colors help you to understand how much you are consuming?	M

Table 7.5. The questions proposed in the "Direct power feedback" group

Energy Goal setting

For this group, users were required to watch the video reporting EGS data gathered from our house model during all day long. After the video, they have to reply to 14 questions (Table 7.6), all with multiple choice answers. Questions 4 and 5 are randomly chosen from three alternatives; also question 6 is chosen randomly. Due to the similarity between the two interfaces, before starting this questions group, a "separation" page was shown to participants, to explain them that the video presented in this group is different from the previous one. This page has been inserted after a preliminary trial of the web-based survey, where users did not always realize that the video was changed.

We expect that most participants:

- understand the goal-setting strategy;
- comprehend whether and when energy consumption increases or exceeds the goal;
- understand why and how the rooms change colors;
- evaluate the utility of such visualization to improve their energy behavior.

Moreover, we ask for suggestions about how the IHD should define the "goal of tomorrow" if the goal of today was (or was not) exceeded; and whether the IHD should reward them when the energy consumption is lower than the goal.

Energy Goal setting	
1. What is the daily energy consumption that must be respected?	M
2. Does the actual daily energy consumption exceed the predefined limit?	M
3. When does the energy consumption increase?	M
4/5. A room is green if…	A
4/5. A room is red if…	A
4/5. A room is orange if…	A
6. In the previous question, what do you mean for "a little"?	A
6. In the previous question, what do you mean for "a lot"?	A
7. Do you think that every room changes its color with the same energy consumption values?	M
8. How do rooms change color?	M
9. Do the green, red and orange colors help you in understanding how you are behaving with respect to your energy goals?	M
10. If today I've met my energy consumption goal, how shall the goal of tomorrow be defined?	M
11. Do you think that the next energy consumption objective shall take in account how much you exceeded the goal for today?	M
12. How do you like to take into account the energy consumption excess?	M
13. Do you think you shall be rewarded when your energy consumption is lower than the daily objective?	M
14. How do you like to be rewarded?	M

Table 7.6. The questions proposed in the "Energy Goal setting" group

Final rush…

In this question group, we asked for suggestions and preferences about the two visualizations and the IHD in general (Table 7.7). Participants have to reply to five questions, in this group. In the end, we asked for general suggestions and preferences about the presence of an IHD in the house. The answers in this group are either free text or multiple choice.

7.5 Results

1807 people participated in the survey. 992 completed the questionnaire while 815 did not, thus the overall completion rate was 54.89%. No follow-up techniques were applied to reduce the amount of non-answering participants. Most of people who did not complete the survey answered the first two groups of questions and started the third, but did not continue presumably because they underestimated the duration

Final rush. . .	
1. With reference to the previous clips, which of the two interfaces would you like to have in your home?	M
2. What interface would motivate you to reduce your energy consumption?	M
3. Would you like to have this screen in your home?	M
4. Where, in your home, would you like to install the screen showing this interface?	M
5. Suggestions? Comments?	O

Table 7.7. The questions proposed in the "Final rush. . . " group

of the survey and decided to interrupt it or because they did not understand the differences between the first and the second video, and therefore refused to provide duplicate answers. Their answers are not part of the results and the discussion reported in this chapter.

The questionnaire is based on an open sample of people and, as such, the results cannot be proven to be representative of any given population. But with nearly 1000 responses collected, "patterns can be identified and cross-discipline analysis is possible" [40].

The majority of people that finished the survey are from academia (76%), with the rest coming from industry. Most academic people are students at Politecnico di Torino (88%) having an educational background mainly focused on engineering, architecture and industrial design. Participants are aged from 18 to 70 years (M: 23, SD: 8.36); 686 (69.15%) are male, while the other 306 are female. Most of the users come from Western countries, in particular: 945 people come from Italy (95.26%), 15 from Spain (1.51%), 8 from Finland (0.81%) and 7 from the United States (0.71%), as reported in Table 7.8.

Country	**# participants**	**% participants**
Italy	945	95.26%
Spain	15	1.51%
Finland	8	0.81%
United States of America	7	0.71%
France	5	0.50%
Others	12	1.2%

Table 7.8. The country of the questionnaire participants

Next subsections will discuss survey results, divided by group at a question-level granularity (see Appendix A for finer details).

7.5.1 Direct Power Feedback

Questions of this group were about instantaneous power consumption visualization (see Table A.1 in Appendix A for more details).

When asked, after watching the first video, *“What could be the maximum power allocation defined for the home in the video?”*, 50.81% of our respondents correctly answered “3 kW”. Since no evidence of this value is reported in the video, the number had to be estimated by looking at the color of the total consumed power. The total power consumption indicator, in the video, becomes red when it reaches 2.67 kW, at 06:02 PM. This behavior suggests that the maximum power could be around 3 kW. For this reason, we consider “reasonably correct” also the reply “2.7 kW”, given by 32.56% of our respondents. The total percentage of correct replies was 83.37% and fits our expectations.

The next question, *“When was the power consumption highest?”* was answered correctly by 92.54% of the participants, who identify the maximum power consumption between 06:00 and 06:05 PM. The same happens with the third question: *“What appliance consumed most power?”*, where 94.56% of our respondents identify the washing machine as the most power consuming appliance. These preliminary replies suggest that almost all the participants understood where to find this information and how to read it.

The next set of questions looks for changes in room colors. Each participant was randomly shown two of the questions: *“A room is green if. . . ”*, *“A room is orange if. . . ”* and *“A room is red if. . . ”*. 71.56% of respondents of the first question answered nearly correctly (“Nothing is on”) while the 26.30% answered correctly “Something is on and it has a low consumption”. Even if the second reply is the best, the former is not totally incorrect since, in the video, rooms are green with no appliances turned on. Things go better with the second question, where 53.87% of the respondents answered correctly “Something is on and it consumes a bit”. A significant portion of the participants (36.68%) answered “Something is on and it has a low consumption”. For our purpose, we also considered correct this answer, mainly to account the ambiguity of the terms “a bit” and “low”. The same ambiguity is much lower for the last question (“A room is red if. . . ”) and this is reflected by the high percentage of correct replies (85.71%). In all questions belonging to this set, respondents perceived the general difference between room coloring, as expected.

The last question of this group asks for an opinion about the color-based visualization: *“Do the red, orange and green colors help you to understand how much you are consuming?”*. 71.77% of respondents answered “Yes” and 25.40% said “A bit”. We imagined an higher number of positive replies for this question but we consider satisfactory the resulting figures, especially if compared with the negative replies (2.82%).

7.5.2 Energy Goal Setting

Questions of this group were about goal-based visualization, on energy consumptions (see Table A.2 in Appendix A for more details). After watching the video, we asked *"What is the daily energy consumption that must be respected?"*. 92.44% of our respondent correctly answered "7 kWh". This value, however, is clearly reported in the video, thus being easy to spot.

The following question, *"Does the actual daily energy consumption exceed the pre-defined limit?"*, was answered correctly by 94.56% of participants. The same happens with the next question: *"When does the energy consumption increase?"*, where 52.82% of our respondents answered "When a new device is switched on" and 44.46% answered "Only if there are active devices". We considered both questions as correct, since the energy consumption increased in both cases. These preliminary replies suggest that almost all the participants understood where to find this information and how to interpret it.

Room Colors

The next set of questions looks for changes in room colors, similar to the previous question group. Each participant has been randomly presented two of the questions: *"A room is green if. . . "*, *"A room is orange if. . . "* and *"A room is red if. . . "*. 78.29% of respondents of the first question answered correctly "Until now, the devices located in the room have consumed a little". More or less, the same happens with the second question, where 60.70% of participants gave the correct answer ("Until now, the devices located in the room have consumed quite a bit"). As for the previous question group, we considered both answers as correct, due to the ambiguity of terms "quite a bit" and "a little". Such ambiguity is really lower for the last question ("A room is red if. . . ") and this fact is confirmed by the higher percentage of correct answers (80.42%). Respondents, for this set of three questions, perceived the general difference in the room coloring, as expected.

To better understand what people mean when choosing "a little" or "a lot", we asked a further question, randomly chosen from *"In the previous question, what do you mean for 'a little'?"* and *"In the previous question, what do you mean for 'a lot'?"*. In the survey design, the two quantifiers ("little" and "a lot") corresponded to respectively less than the 40% of the daily consumption goal for the room, and to more than the 80% of the same goal. Only 43.28% of respondents answered correctly to the first question ("Less than the energy consumption objective associated to the room"). 41.30% answered "Less than 1 kWh". Even if this answer was not totally correct, in the example shown in the video, all the rooms are green when the energy consumption is lower than 1 kWh. This fact, probably, indicates that several users did not totally understand the correct algorithm but they understood the behavior presented in the example, and deduced the answer from the video.

For the second question, 62.47% of our respondents answered correctly "More than the energy consumption objective associated to the room". A significant portion of answers (21.17%) were "More than 3 kWh", again suggesting that these users did not completely understand the correct algorithm but they understood the behavior presented in the video since, for example, the kitchen becomes red when its energy consumption is higher than 3.1 kWh. In our opinion, the percentage of users answering correctly is higher than before because it is easier to mark the alternative answers as "wrong", by observing the behavior of the room coloring in the video.

At this point, we asked participants *"Do you think that every room changes its color with the same energy consumption value?"*. 68.55% of respondents said "No", that is the correct answer. However, 18.04% answered "Yes", while 13.41% said "Maybe". The next question, *"How do rooms change color?"* had again two acceptable answers. Most users chose one of these answers. In particular, 50.40% of respondents chose the most correct answer ("On the basis of the total energy objective referred to a single room"), while 26.71% said "On the basis of the energy consumed until now" that is a little less correct but is not a wrong answer since each room changes its color according to the energy consumed inside it.

The next question of this group asks for an opinion about the color-based visualization: *"Do the red, orange and green colors help you in understanding how much you are behaving with respect to your energy goals?"*. 43.35% of respondents answered "Yes" and 43.55% said "A bit". Such a result is in accordance to our expectations, due to the "complexity" of the EGS visualization, especially if compared with DPF.

Goal of Tomorrow

The last five questions were not directly related to the video but they concerned the "goal setting for tomorrow". How shall it be defined? Shall it take into account how much the user exceeded the goal for today? How? Shall you be rewarded when your consumption is lower than the daily goal? How?

Participants had different ideas. 65.02% of our respondents said that if they met the goal for today, the goal of tomorrow should be lower. This answer could suggest an attempt to improve their personal energy-saving behavior. However, 32.56% of users said that if they met the daily goal, the objective for tomorrow should be equal. When asked *"Do you think that the next energy consumption objective shall take into account how much you exceeded the goal for today?"*, 72.58% said "Yes" and 8.67% answered "Maybe". The 806 respondents that provided positive responses to the previous question, however, did not have convergent opinions on how take in account the energy consumption excess. In fact, 37.27% of them said that the new goal should be decreased with a part of today's excess; 29.81% said that the new goal should be decreased with the entire today's excess; 19.25%, finally, would have increased the new objective by a part of today's energy excess. It is interesting to notice that more than 60% of these respondents would decrease the goal, thus "punishing" themselves to have exceeded the daily quota.

We also collected participants' opinions about a reward to give them if they met (or over-met) their daily goal. 54.13% of our respondents said that they did not want a reward; only 36.59% said "Yes". 33.49% of respondents who asked for a reward said that they would decrement the new goal by a part of the energy saved; 17.09% would decrement the new goal by the entire quota of energy saved today; 16.40% would increment the new objective by a part of the energy saved. 25.28%, however, suggested other rewards. The most popular suggestion was an economical reward, on the final price of the energy bill.

7.5.3 Final rush...

The last question group asked users for opinions, preferences and general suggestions. In the first question, we collected a preference about which interface (DPF and/or EGS) users would like to have in their homes. 47.98% of respondents expressed the desire of having both interfaces, 28.83% chose DPF (the former) and a nearby percentage (21.37%) EGS. Only a small group of persons answered that they would not like to have any interface in their homes (1.81%). This absence of bias between the two interfaces was not preserved when asking participants *"What interface would motivate you more to reduce your energy consumption?"*. 49.90% answered "Goal (energy consumption)" interface and 36.49% the other one. Only a 13.61% said that the two interfaces are equivalent. Even if almost half respondents would like to have both interfaces in their home, a larger subset thought that the EGS visualization could improve their "green behavior" more than DPF.

Next questions referred to whether and where participants would have an IHD in their homes. 37.30% of respondents would have an IHD screen, if possible; 31.25% probably would have; 24.19% think that they absolutely need such a screen; only 7.26% would not have any IHD. Regarding the location of the screen, most users reported more than one room. In particular, the most frequently mentioned room was the kitchen (32.66% of preferences), followed by the lobby/corridor (20.44%). The third preference went to a generic "most popular room" (13.36%). It is interesting to notice that 4.19% of preferences regarded portable devices or integration with pre-existent appliances, but only 1.40% of replies explicitly indicated the "most consuming room" as a good location for such displays. Even the bathroom/laundry collected few preferences (1.66%).

The last question asked participants for comments and suggestions (if any). The most interesting replies are reported and discussed in the next Section.

7.6 Discussion and User Comments

The objectives of our questionnaire were to understand if people like IHDs and comprehend energy goal setting or direct power feedback visualizations. Our survey results indicate that most of our respondents would adopt an in-home display, thus demonstrating a strong motivation to save energy. About half of them would like to have both visualizations in their IHD, but they prefer the energy goal setting one if the final objective is to improve their green behavior. It seems that direct power feedback visualization is more useful for checking the presence of turned on appliances that nobody uses and for avoiding to exceed the maximum power allocation for the home, while energy goal setting is better for improving energy consumption and the personal green behavior. Results also show that color-based feedback is easily understood and well appreciated, especially in the DPF case; moreover, the direct power feedback visualization appears to be easier to understand than the energy goal setting one.

Regarding the location of an IHD in the house, most users suggest to place it in the kitchen or in the lobby. Two trends emerge from the comments gathered by this question: about half of respondents, in choosing a location, looked for a visible and central place, while the others suggested places less visible but “esthetically acceptable,” for example by indicating to put the IHD near the electricity control system (i.e., energy meter and/or circuit breaker). Other users suggested to have the direct power feedback visualization in every room (or on a portable device, such as a PDA, a smartphone or a digital picture frame), and the energy goal setting only in one room, with a dedicated screen. Moreover, the few participants that suggest to put the IHD in the bedroom stress the educational aspect of energy and power saving, especially for their children.

The last question of our web-based survey looks for general comments and suggestions. Omitting the comments about the questionnaire itself (most of them are positive) and the difficulties experienced by some participants in understanding the behavior of the EGS visualization, it is possible to gather suggestions in the following ten sentences, ordered by popularity:

1. report the partial power/energy consumption for each room, also numerically;

2. realize an joint interface for both visualizations;

3. offer the possibility to set a goal not only on a daily base, e.g., weekly;

4. offer a power/energy consumption history;

5. offer control of appliances;

6. add an alarm to report when the circuit breaker is near to be activated;

7. give hints about how to improve current green behavior in both visualizations;
8. provide appliance-level detail for instantaneous power consumption data;
9. take into account, in the EGS visualization, recurrent behaviors and seasonal patterns;
10. give the possibility to set custom goals at the room level.

The most notable concept in this list is that almost all suggestions are about the energy goal setting visualization: only two of them regard solely the direct power feedback interface. The first comment (the most popular) suggests to report the energy and power consumption for each room, not only with colors but also with a numerical value. This request for more details at room level could suggest some difficulties in the color-based visualization whose behavior could be clearer by adding some details about the single room.

The second suggestion is related to the fact that about half of our respondents would like to have both interfaces.

The third comment is about goal duration: users prefer to work with weekly or monthly goal. This option was already considered in the interface design, but does not appear in the shown video: with a weekly goal, the video would have been too long.

Next comments regard possible improvements of our visualizations, to be exploited in future work. The most interesting improvement is the request for a consumption history, to maintain separately for both visualizations (#4) and to integrate with the energy goal setting (#9). The request for hints to improve users energy behavior (#7) and the suggestion to extend the interface by including the control of (smart) appliances (#5), so that users could act on various devices as soon as they see single appliance consumptions in the IHD, are interesting. The last suggestion, in particular, confirms the relevant role of home automation in saving energy at home.

7.7 Conclusions

This chapter presented a web-based questionnaire with the main goal of validating the interaction paradigms of direct power feedback and energy goal setting visualizations against the needs of a wide user base (992 users) of people living in a home, and habitually using basic web technologies such as a web browser or an e-mail-reading program (technology-aware). Results show that most respondents would like to have an IHD in a central place of their home and that they understand and accept both direct feedback and goal setting visualization, even if they feel the latter more useful for reducing their energy consumptions.

Room-level detail proposed by both visualizations proved to be interesting, on one hand, but on the other hand it showed some shortcomings, especially in the goal-setting visualization, where more “precise” (numeric) feedback was required by most of survey respondents. This motivates further research on level-of-detail aspects.

Interesting insights resulted from the question group about the “goal of tomorrow”, i.e., about which policy might be better to set-up the next goal when a goal validity time expires. First it is rather clear that people are not really aware of how to set and modify such a goal, although they are kind to commit to greener behaviors. Second, it is surprising that such a commitment is reflected in setting more stringent goals even when the just-ended one was missed. Monetary rewards still preserve some attraction but most of people participating in the survey would improve their energy efficiency for free.

Chapter 8

WattsUp: Improving Power Consumptions at Home

This chapter stems upon the prerequisites and the results of the User Survey described in the previous chapter. To briefly recap, the assumptions made in the User Survey was:

- the availability of one meter per room or of an equivalent metering system able to provide measurements at room-level granularity;
- the availability of a home automation plant able to detect and report home device activations;
- the availability of a medium-sized (e.g., 7" or greater) display hardware.

To accomplish the third point, an Android app named *WattsUp* has been developed and evaluated on a 10" tablet; it covers the "Direct Power Feedback" part of the survey. The second point is fundamental in a Smart Environment, so it is given by default. Eventually, the first point introduces a difficulty, since almost none home automation plant has one energy meter per room: for this reason, the DogPower ontology and the related bundles in the Dog gateway has been designed and developed.

The current implementation of WattsUp strongly depends on these prerequisites, thus relying upon this ontology and its usage inside Dog.

8.1 Motivation

Energy efficiency has become one of the major concerns in every human activity, impacting on almost all aspects of the human life, from industrial and commercial activities to leisure and holiday. In the last years, moreover, an evolved green consciousness, a higher attention to sustainable development and some worrying forecasts on future energy shortages have fostered renewed attention on more efficient consumption. According to statistics from both the US Department of Energy and the European Union Energy Commission, global consumption will still increase in the next years, with residential and commercial buildings raising their aggregate figure to 20%-40% of the total yearly consumption. If only electricity is considered, the consumption share allocated to buildings raises up to 73%, being almost evenly distributed between residential and commercial buildings [87]. In other words around 1/3 of the total electric energy consumption in the USA and in the EU will be allocated to homes. This high share of energy fosters an ever-increasing momentum of energy efficiency research applied to residential houses, also reflected by current incentive programs issued by both USA and EU governments.

While most efforts in the literature are currently concentrating on single device efficiency, or on local-production systems, the home of the future may exploit the full potential of nowadays home automation to help reducing and rationalizing current energy use in the home. Smart homes can play a pivotal role in the future, by enabling users to better organize their daily activities in order to reduce the global home consumption, by suggesting and promoting new, more efficient behaviors and by preventing or postponing the activation of energy greedy appliances, possibly coordinating with local power sources. Fine grained metering is one of the key factor for these "energy positive" innovations in the homes although implied costs still prevent its application inside home environments. As a consequence, while commercial and industrial dwellings are starting to employ more and more metering solutions to account, monitor and reduce energy wastes along the entire production chain, homes are still "locked" into a stale condition where only one meter (if any) is installed and almost no policy can be applied.

To overcome this issue and start improving energy efficiency of residential habitations we propose to enrich current smart homes with explicit, machine understandable, energy information, in form of appliance-level power consumption data, either nominal or measured on the specific device. In our view, every home device and appliance that can be controlled, i.e., activated, by an home automation plant is modeled to account for its power consumption information. Such a detailed modeling allows to estimate the total power, and, in a given time interval, also the total energy absorbed by a smart home, by knowing device activations, only. If such a capability is complemented by the availability of one, or more, real meters, estimation can be improved and results may increase their accuracy scaling gracefully to the full metering case, where every device is connected to a dedicated meter.

The DogPower ontology model presented in this chapter is specifically designed to model nominal, typical and real power consumption of each device in a home and its modular design allows to plug the same model into different ontology-based modeling frameworks for smart homes. A DogPower empowered intelligent environment is able to estimate its current power consumption just by knowing the current state (on, off, standby, moving, . . .) of connected devices and by consequently querying the ontology power model we propose. Such a capability can be variously exploited and we report four different use cases in which DogPower may successfully increase home energy efficiency.

To support the feasibility of the proposed approach an integration of the DogPower model into the ontology-powered Dog gateway has been provided, laying the basis for energy efficiency support in heterogeneous environments based on commercial technologies, e.g., Konnex, BTicino MyHome and Z-Wave.

Moreover, the WattsUp application, realized as an Android app targeting 10” tablets, provides a practical usage of such an ontology and its integration in the Dog gateway, to fulfill the prerequisites and the results of the user survey presented in the previous chapter.

8.2 Use Cases

Automated homes able to exploit power information (e.g., by leveraging the DogPower ontology described in Section 8.3) have the potential to profoundly impact the energy efficiency of households, promoting greener behaviors and more energy efficient interactions. With respect to current metering-based approaches, DogPower enables energy efficiency policies also in habitations with few or no meters, thus complementing and enriching the possibilities of achieving improvements in the current energy balance of residential houses. To better exemplify how a relatively simple and abstract model of energy consumption such as DogPower and an application like WattsUp can positively impact the energy consumption in buildings, the following subsections depict three different use cases, of increasing complexity, where power modeling assumes a crucial role for sustaining better consumption policies.

8.2.1 Meterless energy monitoring

Many research contributions, in literature, show that householders can decrease their electricity absorption by 5%-15% by just being informed about their current electrical consumption habits [44, 43]. Such a figure can grow even further if householders are highly motivated and if the billing scheme incentives reductions, e.g., prepaid billing where users pay before consuming. To make home inhabitants aware of their current energy habits, most approaches exploit In-Home energy Displays (IHDs) [38, 45, 41], like WattsUp, showing the amount of power currently consumed, the energy absorbed since a given start time and information to promote positive changes in the householders lifestyle. Metering is a functional requirement for IHDs, if no metering information is available in the home, no feedback can be given and no improvement can be achieved. This is true even if one meter is always installed in homes: the one provided by the energy supplier. Such a meter, in fact, is often hidden or not accessible, and the information it provides comes back to the user with a monthly granularity, only. While home automation may be easily bundled (in new homes) or retro-fitted into domestic spaces, metering is seldom available, and even simple meters have relatively high costs which might prevent their installation. If we consider, for example, Z-Wave networks, typical meter prices range from 50 to 100 euros. Given a medium habitation with a set of 5-6 white goods installed and a variable number of brown appliances distributed in all the environment, metering costs can easily raise to not-convenient levels, preventing effective application of energy awareness IHDs. By simply plugging DogPower device descriptions into the existing home automation plant, i.e., by inserting a proper software module into the existing home automation gateway, it is possible to estimate the current home consumption from the activation states of the devices connected to the domotic plant. Even if no meter is installed in the home, consumption estimation can still be carried supporting IHDs and, in perspective, promoting energy savings up to 15%. In the case of partial metering installed in the home, estimations can be corrected and improved thus contributing to further refine achievements obtainable in the zero-meter case.

8.2.2 Improving metering granularity

Whenever at least one metering system is installed in an automated home, modeling device consumptions by means of DogPower allows to achieve increased metering granularity, based on the DogPower model and on the current state of devices controlled by the home automation plant. The DogPower model, in fact, allows to split the metered aggregated consumption values into device level estimated measures. Given such a DogPower model of the environment, two phases identify this use case: learning and on-line consumption disaggregation. In the learning phase, the home automation plant refines the DogPower model by applying (or by monitoring) switch-on and switch-off activations of the devices present in the environment (in different operating conditions). During all these activations, the real consumption is monitored through the metering system (one central meter is sufficient) and the DogPower model of the house is accordingly updated. In the second phase, that lasts until a new learning phase is triggered, the home automation plant measures the current consumption through the metering system and, thanks to the DogPower model, associates the appropriate portion of the aggregate consumption to each device, thus contributing to identify greediest appliances and possible candidates for energy-driven interventions. Such interventions may lead to the substitution of some home equipment with more energy efficient ones (efficiency behaviors [88]) with a long-term effect on the overall energy efficiency of the household.

8.2.3 Better Practices Suggestion

Given an automated home with one or more meters installed, DogPower may be used by the home gateway to implement advanced suggestion policies, stimulating positive changes of the home inhabitants behaviors. At every device activation, e.g., by means of a button, the home intelligence may, in fact, check the alternative ways for achieving the same final home state, and suggest to activate the one having the lowest impact on the home energy consumption. For example, we can imagine that a bathroom shall be illuminated. The home middleware knows that bath illumination can be obtained by either switching on the bathroom ceiling lamp, by turning on the lamp on the top of the bathroom mirror or by raising up the bathroom shutter. Every solution can be profiled under the energy consumption point of view, using information encoded in DogPower, and the less consuming one can be identified. If the user selects the best activation (energetically speaking) no suggestion is given, otherwise the home gateway can exploit IHDs (or text-to-speech interaction, for example) to inform the home inhabitant of the existence of a better habit, e.g., raising the shutter instead of lighting the lamp (or vice-versa depending on the context).

8.3 The DogPower Ontology

DogPower is a light-weight ontology designed to model power consumption of electrical devices and appliances in (automated) homes (see Figure 8.1), supporting the previously depicted use cases. A minimal approach is adopted, reducing modeling primitives (classes and relations) to those strictly needed to support power consumption modeling. Relations to described devices and appliances are left “open,” i.e., their descriptions shall be completely formalized depending on the ontology-based home/device model to which DogPower is connected. We explicitly avoid linking upper-level ontologies such as DOLCE [91] or SUMO [92] to enable designers of device ontologies to freely connect their existent ontologies to DogPower, without any constraints on “imported” or “connected” models.

8.3.1 Structure

Two main classes compose the ontology, namely `dogP:PowerConsumption` and `dogP:PowerConsumptionValue`. They respectively model the kind of power absorbed by a given device or appliance, e.g., electric power or thermal power, and the amount (value) of consumed power in terms of International System units, e.g., Watt.

For the purpose of this chapter, the `dogP:PowerConsumption` concept is specialized into `dogP:ElectricPowerConsumption`, modeling electric power typically

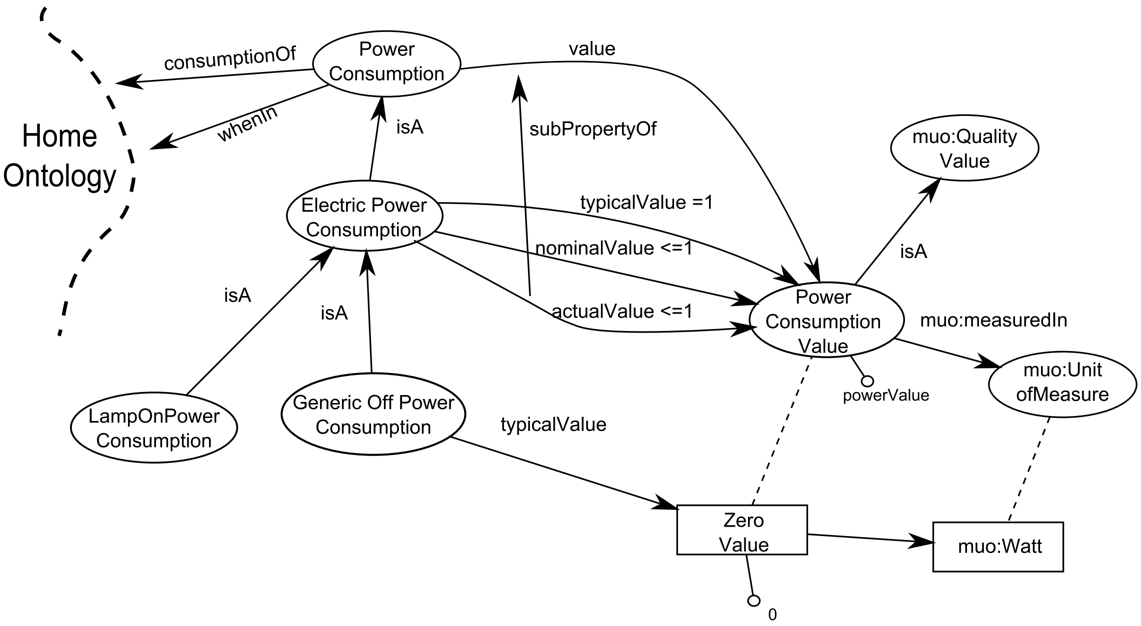


Figure 8.1. DogPower, class hierarchy

absorbed by home devices and appliances. Other types of power consumptions, such as thermal power consumption, can be modeled.

Devices can be described by either instantiating the type of power they absorb or by further specializing the electric power consumption class and by instantiating the corresponding descendants, e.g., `dogP:LampOnPowerConsumption` that models power absorbed by switched-on lamps. Power values are referred to their unit of measure by means of the `muo:measuredIn` relation, defined in the standard MUO ontology [93], that relates a `dogP:PowerConsumptionValue` instance to a `muo:UnitOfMeasure` instance.

Power consumption classes are related to (at least) one power consumption value by means of the `dogP:value` relation. Such a relation is further specialized to allow specifying typical, nominal and actual consumption values. Given an instance of a class descending from `dogP:PowerConsumption`, e.g., of `LampOnPowerConsumption`, such an instance can be connected to:

(a) a *typical* power value derived from the consumption of a generic device of the same type, e.g., lamps in a home typically consume around 60/90 Watt;

(b) an optional *nominal* power value declared by the device manufacturer, e.g., the living room lamp has a nominal power consumption of 30 Watt;

(c) an optional *actual* power value measured on the real device, e.g., the living room lamp actually consumes 28,92 Watt.

Two "open" relations, respectively named `dogP:consumptionOf` and `dogP:whenIn`, model the power absorbed by a given device in a given operating condition. The former relates a device, expressed in whatever ontology (range unspecified) with its corresponding power consumption described in DogPower. The latter further specializes this relation by associating the power consumption with a specific device operating condition, e.g., state; also in this case the relation is left open to achieve maximum modeling flexibility (range unspecified).

Example 1 Consider a sample fluorescent lamp installed in a typical home. General knowledge about lamps[1] allows deriving the typical power consumption of fluorescent lamps used in houses, in a given country, e.g., 18 W. However, since the sample lamp is a real object with nominal working parameters we can model the fact that our sample lamp is actually a 9 W fluorescent lamp (nominal value). Once the lamp is connected to a sample smart home equipped with an energy meter, we can also measure the actual power absorbed by the lamp during its various working conditions, e.g., off (0 W) and on (8.57 W). The resulting DogPower model is shown in Figure 8.2 where the device is only roughly modeled since we are concentrating on DogPower modeling, only.

8.3.2 Integration with DogOnt

To better clarify power consumption modeling through DogPower we consider a specific integration sample, where DogPower is integrated with the DogOnt ontology model.

DogPower in DogOnt

Integrating DogPower and DogOnt means exploiting DogOnt concepts as the ranges of `dogP:consumptionOf` and of `dogP:whenIn`. In DogOnt, devices that can be connected to a domotic plant, i.e., controlled, are modeled by the concept `dogont:Controllable`. Therefore, DogPower in DogOnt will specialize the `dogP:consumptionOf` range to `dogont:Controllable`.

Moreover, since in DogOnt different device operating conditions are explicitly modeled by means of concepts belonging to the `dogont:StateValue` hierarchy, the DogPower `dogP:whenIn` relation range will be set at `dogont:StateValue`.

The resulting integration is reported in Figure 8.3 where DogOnt concepts are reported in bold while DogPower concepts are reported in italic font. It is important

[1] e.g., deriving from web sites such as `http://www.seattle.gov/light/conserve` (last visited on January, 2014)

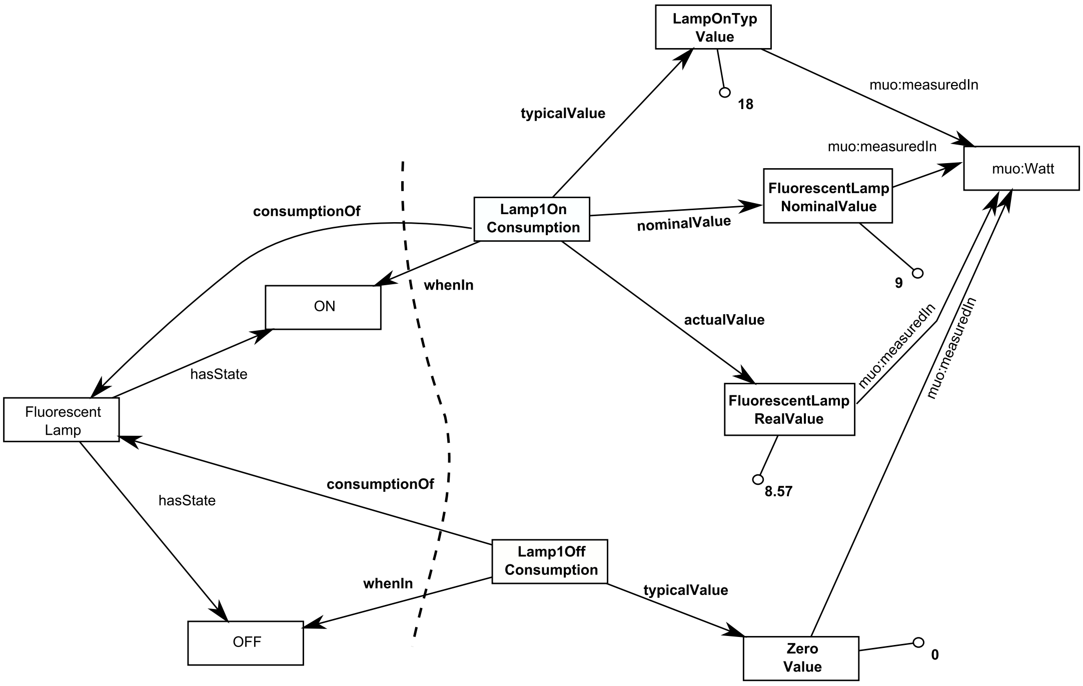


Figure 8.2. A sample DogPower instantiation for a fluorescent lamp

to notice that typical power consumption values can be "predefined" (by means of `owl:hasValue` constraints) for specific consumption classes, e.g., `LampOnPowerConsumption`.

Example 2 Consider a shutter actuator. Thanks to DogPower the actuator power consumption can be modeled and can take part in complex reasoning carried by the "home intelligence". Each of its defined state values is associated to a specific typical consumption, allowing to address "typical" actuators without further knowledge. More accurate power information can also be defined for the given actuator instance (e.g., nominal power, actual power) thus allowing for more precise DogPower models and, in perspective, for more effective policies based on them. Figure 8.4 reports the case in which only typical values are available and only some states are valorized, to show a graph as clear as possible.

8.4 Example Uses

We can use the first and the third use case presented in Sections *Meterless energy monitoring* and *Better Practices Suggestion* as a practical example to exploit some

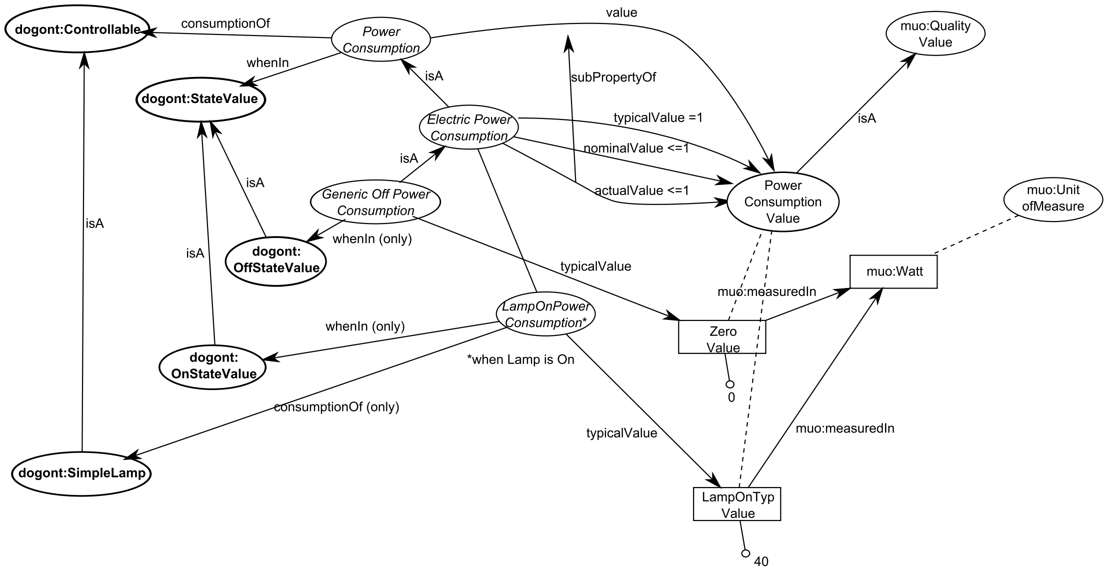


Figure 8.3. Integration of DogPower and DogOnt

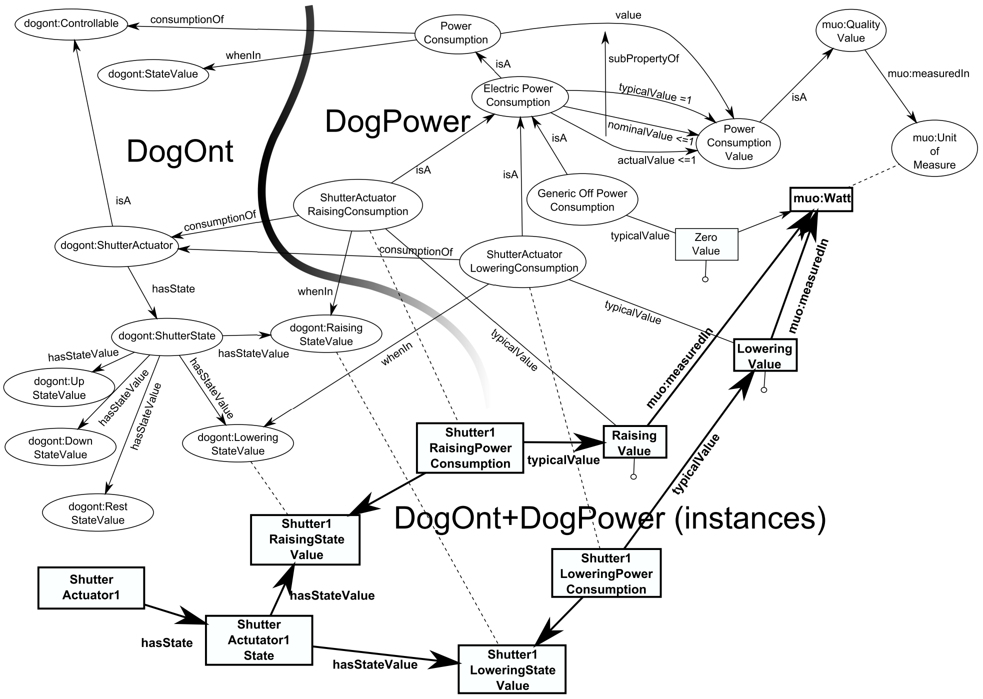


Figure 8.4. A sample shutter actuator model using DogOnt and DogPower

DogPower functionalites. The example environment, modeled using DogPower integrated with DogOnt, could be an home equipped with some smart meters (that do not meter all the home appliances) and automated devices. In particular, we can consider the devices present in the bathroom, as in Section 8.2.3: a lamp on the top of the bathroom mirror, a shutter, and the ceiling lamp. Only the shutter is metered.

If we want to obtain the consumption of such home appliances (metered or not), in order to promote energy saving, we could use a SPARQL query, thus reporting either the typical, nominal, or measured value. As shown in Figure 8.5, SPARQL querying is exploited to extract, for each controlled device `?device` located in the bathroom, the list of its state values `?stateValue` and to further retrieve the power consumption values `?consumption` for each state value. Then, the typical consumption value (`?typicalConsumptionValue`), the nominal (`?nominalConsumptionValue`) and the measured one (`?actualConsumptionValue`) are extracted, if available, from `?consumption`.

```
SELECT ?device, ?typicalConsumptionValue, ?nominalConsumptionValue, ?measuredConsumptionValue
WHERE
{
?device a dogOnt:Controllable.
?device dogOnt:isIn <http://elite.polito.it/ontologies/samples/samplePower.owl#Bathroom>.
?consumption a dogPower:ElectricPowerConsumption.
?device dogOnt:hasState ?state.
?state dogOnt:hasStateValue ?stateValue.
?consumption dogPower:consumptionOf ?device.
?consumption dogPower:whenIn ?stateValue.
OPTIONAL{?consumption dogPower:typicalValue ?typicalConsumptionValue}.
OPTIONAL{?consumption dogPower:nominalValue ?nominalConsumptionValue}.
OPTIONAL{?consumption dogPower:actualValue ?measuredConsumptionValue}
}
```

Figure 8.5. SPARQL query for the “Monitoring” case

By considering again the three devices present in the bathroom, we can imagine suggesting to home inhabitants what is the least power consuming device needed to “illuminate” the bathroom, for example. In fact, we can compare the power consumption of the bathroom devices, taking into account the most precise value for each device, and thus giving inhabitants the indication about what is the least consuming device in the bathroom needed to reach the desired illumination. Such a device can be retrieved by using the SPARQL query shown in Figure 8.6, where the three `OPTIONAL` statements give the more precise power consumption, i.e., if the measured value is available, it is given as `?consumptionValue`; if it is not available, `?consumptionValue` stores the nominal consumption; while if neither is available, then the typical value is reported in `?consumptionValue`. In this way, a home gateway can exploit an user interface to inform the home inhabitant about the existence of a better habit, e.g., raising the shutter instead of lighting the lamp (or vice-versa).

```
SELECT ?device
WHERE
{
?device a dogOnt:Controllable.
?device dogOnt:isIn <http://elite.polito.it/ontologies/samples/samplePower.owl#Bathroom>.
?consumption a dogPower:ElectricPowerConsumption.
?device dogOnt:hasState ?state.
?state dogOnt:hasStateValue ?stateValue.
?consumption dogPower:consumptionOf ?device.
?consumption dogPower:whenIn ?stateValue.
OPTIONAL{?consumption dogPower:actualValue ?consumptionValue}.
OPTIONAL{?consumption dogPower:nominalValue ?consumptionValue}.
OPTIONAL{?consumption dogPower:typicalValue ?consumptionValue}
}
ORDER BY ASC(?consumptionValue)
LIMIT 1
```

Figure 8.6. SPARQL query for the “Suggestion” case

8.5 DogPower Integration in Dog

To realize the prerequisite of the User Survey reported in the previous chapter and to effectively explore the possibilities offered by the DogPower ontology, an integration in the Dog gateway has been performed by creating two separate bundles: Power Model and Power Manager.

The **Power Model** bundle provides an “extension” to the *Semantic House Model*, by adding the power-related information to the abstract device model present inside Dog: the DogPower ontology is now used as an extension of DogOnt and, similarly, the Power Model is an extension of the Semantic House Model (that uses DogOnt, ndr). Such a bundle offers to the running OSGi framework all the service needed to realize the previously described use cases and examples: it can provide the “best” available power estimation, according to the specified nominal, typical and actual values; or it can communicate the overall consumption of a set of devices, for example.

The **Power Manager** bundle consumes the services provided by the Power Model, by adding the proper power estimation value every time a controlled home appliance changes its state (e.g., when a lamp turns off, its power estimation is set to zero) and by providing some power notifications for all devices.

Moreover, the Power Manager bundle is also able to improve the power estimation given by the Power Model bundle if a real electricity meter is present in the home. In fact, in Dog and DogOnt, each meter is declared as “connected to” one or more device (and viceversa, each device can be connected to a meter). If we consider the case where multiple devices are connected to a single meter, the bundle is able to associate a portion of such a “real” consumption value to each device, according to its best value in the DogPower ontology (i.e., nominal, typical, or actual), thus providing a more accurate estimation of the effective power value.

Currently, this operation is handled in the simplest possible way (i.e., by performing a linear repartition between the real, overall, power value and the estimated ones) but can be further improved and, possibly, persistently memorized in Dog for successive usage and refinements.

8.6 The WattsUp application

DogPower and its integration with Dog gives the functional requisites to realize the WattsUp application. The design of WattsUp starts from the “Direct Power Feedback” interface proposed, and partially validated, in the user survey reported in the previous chapter.

Even if the survey results highlight the user will to adopt a system similar to the presented one, with both the “Direct Power Feedback” and the “Energy Goal” interfaces, the former one is better understood. For this reason, WattsUp provides only the power-related visualization and functionalities, using exactly the same concepts and findings emerged from the survey.

The final appearance of WattsUp is shown in Figure 8.7. The application is localized in English and two areas can be distinguished:

- *House Consumption* area, in the left part of the application. It strongly resembles the color-based visualization proposed in the user survey and shows the various rooms present in the house in *green* for rooms with low consumptions, *red* for rooms with high consumptions, and *orange* for rooms with moderate consumptions (included between the low and high range established for the green and red colors, respectively).

- *Information* area, in the right part of the application, showing a semaphore representing the overall “green behavior” of the house, the overall instant power consumption, and the most consuming device currently active.

In WattsUp, the change limits between room colors (e.g., from green to red) can be set by acting on two values, alpha and beta, used in combination with the maximum available power load in the formulas 7.1, 7.2 and 7.3 (see the previous chapter for further details). Default values for alpha is 0.2, for beta is 0.8 and for the maximum available power load is 3.0 kW.

The current version of WattsUp consists of only one page: by “tapping” on a room in the *House Consumption* area, it is possible to see all the devices present in the room (with their details and commands they accept) and the overall power consumption of the selected room (Figure 8.8).

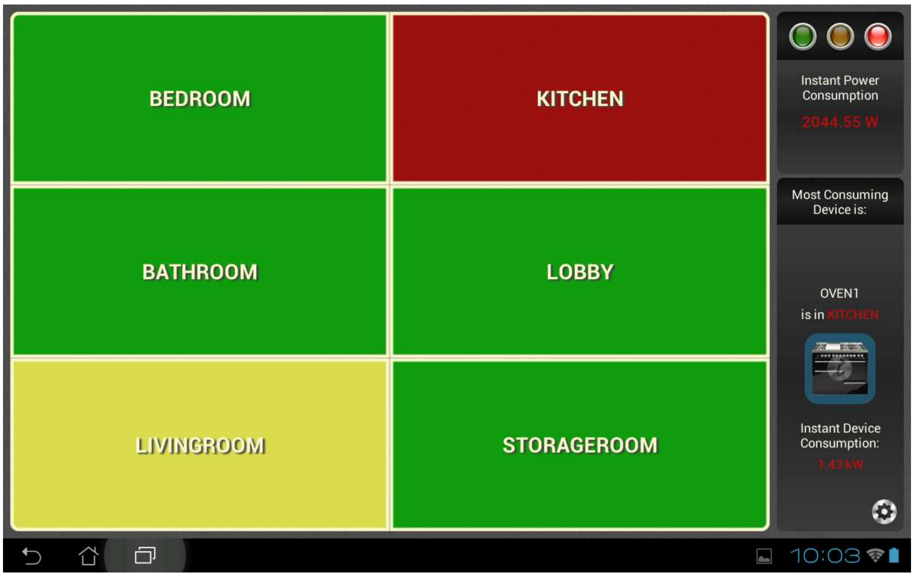


Figure 8.7. The WattsUp application

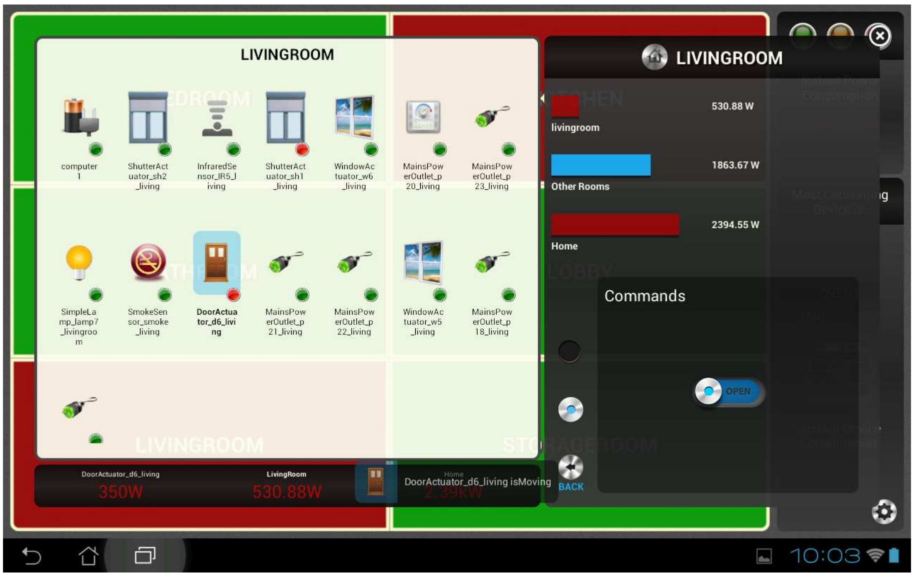

LIVINGROOM
computer1
ShutterActuator_sh2_living
InfraredSensor_IR5_living
ShutterActuator_sh1_living
WindowActuator_w6_living
MainsPowerOutlet_p20_living
MainsPowerOutlet_p23_living
SimpleLamp_lamp7_livingroom
SmokeSensor_smoke_living
DoorActuator_d6_living
MainsPowerOutlet_p21_living
MainsPowerOutlet_p22_living
WindowActuator_w5_living
MainsPowerOutlet_p18_living
LIVINGROOM
530.88 W
livingroom
1863.67 W
Other Rooms
2394.55 W
Home
Commands
OPEN
BACK
DoorActuator_d6_living
350W
LivingRoom
530.88W
DoorActuator_d6_living isMoving
10:03

Figure 8.8. Room Visualization in WattsUp

8.6.1 Implementation

WattsUp has been realized as an Android 4.x application, specifically targeting 10" tablets, for tacking the requirement of a "*medium-sized (e.g., 7" or greater) display hardware*".

The app is connected with the Dog gateway through the REST API offered by the gateway itself, performing command operations and updating the device statuses by polling Dog each second, thus offering up-to-date information. All graphical elements has been customized with respect to the standard Android component, to provide a better look-and-feel and to recall the interface presented in the survey.

WattsUp was tested on an *Asus Transformer Pad*, running Android 4.1 at the moment of the tests.

8.7 Evaluation

WattsUp has been evaluated with a small group of volunteers, to understand its simplicity of usage and to have a feedback about the comprehension of the visualized data, thus initially confirming or denying the results previously obtained.

Six participants used WattsUp in a controlled environment, performing nine tasks each. Test tasks represents common operations that a user need to perform in using the application, for providing information about the easiness in finding and understanding power-related values. Participants never met during the evaluation. Their observation were used to allow qualitative analysis and to help identify strong and weak points of the interface.

Test Deployment A total of six subjects volunteered to participate in this evaluation, with ages ranging from 20 to 30 years, all males. Participants had a variety of educational backgrounds. They were required to have prior general exposure to smartphones or tablets.

After a short introduction to the test and the collection of demographic data, each participant could freely use WattsUp for few minutes. Each participant, then, was told to complete a set of nine tasks, in different order, one at time. For all of them, the *think-aloud* protocol has been adopted, to verify their actions and mental processes adopted. Examples of proposed tasks includes “Identify the third most consuming device”.

At the end, participants were given a questionnaire and asked to rate WattsUp in general and by querying their agreement level (from “Strongly disagree” to “Strongly agree”) about these sentences:

1. I think the application is intuitive.

2. I think that the application layout is understandable.

3. It was easy to learn how to use the application.

Users open comments or explanations were collected (e.g., difficulties found or explanation about something done during a task) through debriefing interviews. The duration of the entire experiment ranged between 15 and 20 minutes.

8.8 Preliminary Results

The final questionnaire asked participants to give an overall grade to WattsUp, in a scale from 1 (the worst) to 5 (the best). Results showed that they were satisfied from the app, with a mean value of 4.

In the test conclusion, users were also asked to express their agreement about the three sentences previously reported. Results were quite satisfying: all the users agree or strongly agree with the proposed sentences about WattsUp.

Moreover, all the participants completed with success all the nine tasks. In some occasions, the tasks execution order (different between participants) has facilitated some operations. Most important, none of them perceived as "not real" the estimated power values assigned to the different devices. When asked, they confirm that they perceive the truthfulness of the visualized data. Finally, during the debriefing sessions, all the users express the usefulness of such an application and the desire to have something similar in their home, thus confirming the survey results.

8.9 Conclusions

In this chapter I introduce DogPower, an ontology based power consumption model for smart environments, and its integration with a middleware, i.e., the Dog gateway. The ontology allows Dog to provide, for each controlled home appliance, either a typical power consumption, a nominal or a measured consumption.

DogPower exploits a minimal approach, by reducing modeling primitives (classes and relations) to those strictly needed to support power consumption modeling. Relations to describe devices and appliances are left "open", i.e., their descriptions shall be completely formalized depending on the ontology-based home/device model to which DogPower is connected, to enable designers of device ontologies to freely connect their existing ontologies to DogPower, without any constraints on upper models.

Moreover, the chapter introduces WattsUp, an Android application targeting 10" tablets that implements the "Direct Power Feedback" interface of the user survey presented in the previous chapter, and uses DogPower and Dog as a way to accomplish the survey prerequisites. Preliminary results from a user evaluation with 6 people seem to confirm the usefulness of the application and the acceptance of the data coming from the overall system.

Chapter 9

Conclusions and Future Work

The previous chapters have outlined a well-defined path. We started with the goal of improving users' interactions with Smart Environments, exploring fundamental principles and issues of such a field, such as its aims of using a *user centered approach* or to consider the user at the center of the overall system, and always in control.

We then discuss a series of approaches, each pursuing a different approach for improving the interaction between the human and the "intelligent" environment, mainly targeting the home. Although each chapter has offered insights that may stand alone, I believe the concepts, design recommendation, prerequisites and requirements, study results, and technical advances are synergistic and can help moving forward for an effective interaction with such environments equipped with an intelligence of their own and, most importantly, really different from the human one.

To conclude this thesis, I now summarize key contributions and possible future works.

9.1 Summary of Contributions

Above all, this dissertation contributes with three different, but complementary, points of view toward an holistic and appropriate interaction with Smart Environments: lowering access barriers, making people aware of what happens in their surroundings, and making the home a "smarter" and more "personalized" place. A baseline can be set, taking together the solutions proposed in these three contribution areas.

DOGeye introduces a convenient access point to the home for those people that cannot interact with the environment in other ways, particularly targeting people with severe motor impairments (e.g., people with ALS). Such persons, at a certain point of their disease, are able to use the eyes as their unique mean of communication and interaction. The contribution of DOGeye is, therefore, two-fold: from one side, it explicits and implements an established set of requirements for eye-tracking applications oriented to home control, presenting the design of a multimodal application with a working implementation; and, from the other side, it shows a modular nature that permits its usage with different smart home systems by applying minor modifications, only.

WristHome shares with DOGeye the objective of lowering the access barriers to SmE. However, it follows an unobtrusive approach to smart home management: while DOGeye targets people with severe motor impairments and proposes a eye-based graphical user interface, WristHome introduces a wearable platform, almost "invisible". There are, in fact, situations or people where is not possible, nor appropriate or secure, to employ devices heavily present in typical home automation systems (and Smart Homes), like displays, computers, or smartphones. For example, the elderly, or people with a low acceptance of technology, would not be forced to bring a smartphone with them, at all times, to interact with their homes, or learn how to use a computer. WristHome employs a wrist watch as an interaction mean, an object that is already present and accepted in most people's life, thus improving interactions and acceptance of an "intelligent" system. Its contribution consists of a set of requirements extrapolated from the literature on wearable and ubiquitous computing, and of a general architecture that realize such requirements. The WristHome prototype is loosely coupled with the middleware used in this thesis, thus allowing its usage in different systems.

RulesBook aims at making the home a "smarter" and more "personalized" place, by letting people to decide what level of autonomy and what tasks should be performed by the Smart Environment. In this way, the user is always in the loop, and perceive the Smart Environment as something convenient and useful, not to be turned off. By delegating repetitive tasks, or composing context-aware applications, the user maintain the control of her house and, at same time, efficiently uses the technology to live better. This is realized with a mobile application, targeting end-users, that employs a formal representation grammar based upon some fixed keyword (`IF`, `WHEN`, `THEN`) to express rules based on a triggered event, one or more constraints and one or more possibile action to perform. In this way, it is possible to delegate to the Smart Home tasks like "*IF a window in the kitchen has been opened, WHEN the heating system is on, THEN turn the kitchen radiator off*". RulesBook contributes with some requirements, obtained both from the literature and from people living and managing smart and automated homes, with the definition of a formal yet extensible grammar for rules composition, and with the design of an application implementing such a grammar. It is totally independent from the underlying middleware, provided the capabilities to successfully handle the established grammar.

Eventually, **WattsUp** tries to promote the education toward a "greener" behavior in the usage of energy in the home. WattsUp presents multiple contributions. First of all, it stems its prerequisites and requirements from a user survey, distributed online between September 2010 and January 2011, and completed by 992 people, for collecting an evaluation about two possible visualization paradigms in the design of an effective user feedback display on energy consumption. Secondly, it presents an ontology model, named *DogPower*, to overcome physical and economical problems such as the typical lack of a room-level metering system in the home; such an ontology has been realized with a modular design to allow the model to be plugged into different ontology-based middleware (a technology that is quite spread in the SmE field). Finally, it introduces an Android application that implements the requirements and realizes the interface designed in the user survey. Even in this case, the application is loosely coupled with the smart system adopted in this dissertation, thus maintaining the findings and the introduced requirements general enough.

9.2 Future Work

All the approaches proposed in this work have been tested with existing evaluation techniques, arising from the Human-Computer Interaction field. However, systems to support everyday life are difficult to evaluate with such "classical" techniques, i.e., in the lab [2]. As this type of interactive applications pervades the life of their users, effectiveness and efficiency, dominant criteria in evaluating traditional computer interface, become less important. Issues of unobtrusiveness, playfulness, enjoyability, acceptance, stability, and autonomy are increasingly important for these interfaces. An effective evaluation will only become possible, and fully significant, when such applications are integrated into the everyday environment.

One possible, and important, future work is therefore an extensive evaluation of such applications "in the field", possibly by employing the middleware already available in the Smart Environment, thanks to the modular architecture of such tools and their loosely coupled nature.

For all the three cited areas, more exploration is possible and desirable. For lowering access barriers, an exploration of possible and useful interaction by using sensing and actuating devices already available in the environment has yet to be extensively performed. For example, lighting system could provide a viable and unobtrusive output mean in several conditions, e.g., by changing the light intensity or its color.

If we consider the area related to make the home a "smarter" and more personalized place, instead, different research activities are still carried on, especially on tools for end-user creation of context-aware applications, such as in the paper proposed by Lee et al. in 2013 [23] or in the paper of Ur et al.[1] to appear at CHI 2014. Further investigation in the area are, obviously, possible and recommended.

An area open for future exploration is a specialization of Wearable Computing: **On-Body Interaction**. The approach proposed by WristHome has been introduced for lowering access barriers and increasing acceptance and unobtrusiveness of a SmE with an always-on and common tool for interfacing with the environment. However, wearable devices suffer from two main drawbacks: they have a limited area for graphical output and for direct input. Graphical output is, nowadays, the most predominant and diffuse means for the communication between the human and a computer system. Direct input, from the other side, suffers from the reduced physical dimensions of buttons and screens available on wearable devices.

On-body interaction aims at overcoming problems related to input and output on small devices by moving them to the human body. This is possible by projecting graphical interfaces on the body (or nearby) and capturing the input through computer vision and bio-acusting techniques. Merging this new field into Smart Environment allows to create a new type of interfaces, that may appear only when needed and that could be able to take into account specific needs of the person how carries the device. For example, by pointing a lamp with an arm, a suitable interface can be projected for turning the lamp on or off; or the environment can be set to send specific bio-feedbacks (e.g., haptic) according to what is happening in the surrounding, thus alerting users of various events without necessarily require their complete attention.

Finally, another important area of future work consists in combining the different approaches followed in this thesis, since they target the same environment (i.e., the home). For example, the approaches employed in WristHome and WattsUp can be combined, to provide simple suggestions or important warnings about energy consumptions directly on a wrist watch, instead of on a larger display, maybe taking into account the location of the user or her current activity for giving the best advice at the right time. The resulting ensemble could be superior to any approach taken alone, yielding a more complete experience.

[1] see `http://www.blaseur.com/papers/TriggerActionCHI14.pdf` for the full paper (last visited on February, 2014)

Bibliography

[1] Diane Cook and Sajal Das. *Smart Environments: Technology, Protocols and Applications*, volume 43 of *Wiley Series on Parallel and Distributed Computing.* Wiley-Interscience, 2004.

[2] Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto. *Handbook of Ambient Intelligence and Smart Environments.* Springer Publishing Company, Incorporated, 1st edition, 2009.

[3] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, 1999.

[4] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2 edition, 2003.

[5] Juan Augusto, Vic Callaghan, Diane Cook, Achilles Kameas, and Ichiro Satoh. Intelligent environments: a manifesto. *Human-centric Computing and Information Sciences*, 3(1):12, 2013.

[6] K Ducatel, M Bogdanowicz, F Scapolo, J Leijten, and J Burgelman. Scenarios for ambient intelligence in 2010. Technical report, ISTAG, 2001.

[7] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, August 2009.

[8] D. Bonino and F. Corno. What would you ask to your home if it were intelligent? Exploring user expectations about next-generation homes. *Journal of Ambient Intelligence and Smart Environments*, 3(2):111–126, January 2011.

[9] I. Mavrommati and J. Darzentas. An overview of ami from a user centered design perspective. In *Intelligent Environments, 2006. IE 06. 2nd IET International Conference on*, volume 2, pages 81–88, July 2006.

[10] Vic Callaghan, Graham Clarke, and Jeannette Chin. Some socio-technical aspects of intelligent buildings and pervasive computing research. *Intelligent Buildings International Journal*, 1:19, 2009.

[11] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, November 1997.

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

[13] F. Corno, E. Castellina, R. Bates, P. Majaranta, H. Istance, and M. Donegan. D2.5 Draft standards for gaze based environmental control. Technical report, COGAIN, 2007.

[14] Tsogzolmaa Saizmaa and Hee-Cheol Kim. A Holistic Understanding of HCI Perspectives on Smart Home. In *NCM '08: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, pages 59–65, Washington, DC, USA, 2008. IEEE Computer Society.

[15] Seong Lee, Ilseok Ko, and Min Kil. A user interface for controlling information appliances in smart homes. In Ngoc Nguyen, Adam Grzech, Robert Howlett, and Lakhmi Jain, editors, *Agent and Multi-Agent Systems: Technologies and Applications*, volume 4496 of *Lecture Notes in Computer Science*, chapter 92, pages 875–883. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2007.

[16] Elisabetta Farella, Mirko Falavigna, and Bruno Riccò. Aware and smart environments: The casattenta project. *Microelectron. J.*, 41(11):697–702, 2010.

[17] Florian Weingarten, Marco Blumendorf, and Sahin Albayrak. Towards multimodal interaction in smart home environments: the home operating system. In *DIS '10: Proceedings of the 8th ACM Conference on Designing Interactive Systems*, pages 430–433, New York, NY, USA, 2010. ACM.

[18] Diane J. Cook and Wenzhan Song. Ambient intelligence and wearable computing: Sensors on the body, in the home, and beyond. *J. Ambient Intell. Smart Environ.*, 1(2):83–86, 2009.

[19] M. T. Raghunath and Chandra Narayanaswami. User interfaces for applications on a wrist watch. *Personal and Ubiquitous Computing*, 6(1):17–30, January 2002.

[20] Brian D. Mayton, Nan Zhao, Matt Aldrich, Nicholas Gillian, and Joseph A. Paradiso. WristQue: A personal sensor wristband. In *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, pages 1–6, May 2013.

[21] Uwe Maurer, Anthony Rowe, Asim Smailagic, and Daniel Siewiorek. Location and Activity Recognition Using eWatch: A Wearable Sensor Platform. In Yang Cai and Julio Abascal, editors, *Ambient Intelligence in Everyday Life*, volume 3864 of *Lecture Notes in Computer Science*, pages 86–102. Springer Berlin / Heidelberg, 2006.

[22] M. T. Raghunath and Chandra Narayanaswami. User Interfaces for Applications on a Wrist Watch. *Personal and Ubiquitous Computing*, 6:17–30, 2002.

[23] Jisoo Lee, Luis Garduño, Erin Walker, and Winslow Burleson. A tangible programming tool for creation of context-aware applications. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous*

Computing, UbiComp '13, pages 391–400, New York, NY, USA, 2013. ACM.

[24] Ricarose Vallarta Roque. *OpenBlocks : an extendable framework for graphical block programming systems.* Massachusetts Institute of Technology, 2007.

[25] Anind Dey, Timothy Sohn, Sara Streng, and Justin Kodama. iCAP: Interactive Prototyping of Context-Aware Applications. In Kenneth Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing*, volume 3968 of *Lecture Notes in Computer Science*, pages 254–271. Springer Berlin / Heidelberg, 2006.

[26] Malcolm Attard and Matthew Montebello. DoNet: a semantic domotic framework. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 997–998, New York, NY, USA, 2006. ACM.

[27] Tom Rodden, Andy Crabtree, Terry Hemmings, Boriana Koleva, Jan Humble, Karl-Petter AAkesson, and Pär Hansson. Between the dazzle of a new building and its eventual corpse: assembling the ubiquitous home. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '04, pages 71–80, New York, NY, USA, 2004. ACM.

[28] Rune Vinther Andersen, Jørn Toftum, Klaus Kaae Andersen, and Bjarne W. Olesen. Survey of occupant behaviour and control of indoor environment in danish dwellings. *Energy and Buildings*, 41(1):11 – 16, 2009.

[29] Ardeshir Mahdavi, Lyudmila Lambeva, Abdolazim Mohammadi, Elham Kabir, and Claus Proglhof. Two case studies on user interactions with buildings' environmental systems. *Bauphysik*, 29:72–75, 2007.

[30] C. F. Reinhart. Lightswitch-2002: a model for manual and automated control of electric lighting and blinds. *Solar Energy*, 77:15–28, 2004.

[31] J. Fergus Nicol. Characterisizing Occupant Behavior in Buildings: Towards a Stochastic Model of Occupant Use of Windows, Lights, Blinds, Heaters and Fans. In *Seventh International IBPSA Conference*, 2001.

[32] D. Bourgeois. *Detailed occupancy prediction, occupancy-sensing control and advanced behavioural modelling within whole-building energy simulation.* PhD thesis, Université Laval - Ville de Québec (Québec), 2005.

[33] Nelson Fumo, Pedro Mago, and Rogelio Luck. Methodology to estimate building energy consumption using energyplus benchmark models. *Energy and Buildings*, 42(12):2331 – 2337, 2010.

[34] Melek Yalcintas. Energy-savings predictions for building-equipment retrofits. *Energy and Buildings*, 40(12):2111 – 2120, 2008.

[35] Haris Doukas, Konstantinos D. Patlitzianas, Konstantinos Iatropoulos, and John Psarras. Intelligent building energy management system using rule sets. *Building and Environment*, 42(10):3562 – 3569, 2007.

[36] Erik Hinterbichler. Designing a better energy consumption indicator interface for the home. Master's thesis, Dartmouth College, 2006.

[37] Tae-Jung Yun. Investigating the impact of a minimalist in-home energy consumption display. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI '09, pages 4417–4422, New York, NY, USA, 2009. ACM.

[38] Helen Ai He and Saul Greenberg. Motivating Sustainable Energy Consumption in the Home. In *ACM CHI Workshop on Defining the Role of HCI in the challenges of Sustainability*, 2009.

[39] James Pierce, Diane J. Schiano, and Eric Paulos. Home, habits, and energy: examining domestic interactions and energy consumption. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1985–1994, New York, NY, USA, 2010. ACM.

[40] M. Pilgrim, N. Bouchlaghem, D. Loveday, and M. Holmes. Towards the efficient use of simulation in building performance analysis: a user survey. *Building Service Emergency Resources Technology*, 24 (3):149–162, 2003.

[41] Simon Roberts, Helen Humphries, and Verity Hyldon. Consumer preferences for improving energy consumption feedback. Technical report, Centre for Sustainable Energy, 2004.

[42] G. Wood and M. Newborough. Energy-use Information Transfer for Intelligent Homes: Enabling Energy Conservation with Central and Local Displays. *Energy and Buildings*, 39(4):495 – 503, 2007.

[43] Ahmad Faruqui, Sanem Sergici, and Ahmed Sharif. The impact of informational feedback on energy consumption – a survey of the experimental evidence. *Energy*, 35(4):1598 – 1608, 2010.

[44] Sarah Darby. The Effectiveness of Feedback on Energy Consumption. Technical report, Environmental Change Institute, University of Oxford, 2006.

[45] Yann Riche, Jonathan Dodge, and Ronald A. Metoyer. Studying always-on electricity feedback in the home. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1995–1998, New York, NY, USA, 2010. ACM.

[46] T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies*, 43(5-6):907–928, 1995.

[47] Dario Bonino and Fulvio Corno. DogOnt - Ontology Modeling for Intelligent Domotic Environments. In Amith Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *International Semantic Web Conference*, number 5318 in LNCS, pages 790–803. Springer-Verlag, October 2008.

[48] The OSGi Alliance. OSGi Release 5. `http://www.osgi.org/Specifications`, 2012.

[49] Fulvio Corno and Dario Bonino. spChains: A Declarative Framework for

Data Stream Processing in Pervasive Applications. *Procedia Computer Science*, 10C:316–323, 2012.

[50] I. Bierhoff and et. al. Smart Home Environment. Technical report, Towards an Inclusive Future, COST219 Project, 2007.

[51] A. van Berlo. Smart home technology: Have older people paved the way? *Gerontechnology*, 2(1):77–87, September 2002.

[52] Bin Zhang, Pei-Luen Patrick Rau, and Gavriel Salvendy. Design and evaluation of smart home user interface: effects of age, tasks and intelligence level. *Behav. Inf. Technol.*, 28(3):239–249, 2009.

[53] Belkacem Chikhaoui and Hélène Pigot. Towards analytical evaluation of human machine interfaces developed in the context of smart homes. *Interacting with Computers*, 22:449—-464, 2010.

[54] Marie Chan, Eric Campo, Daniel Estève, and Jean-Yves Fourniols. Smart homes – current features and future perspectives. *Maturitas*, 64(2):90–97, 2009.

[55] Anthony J. Hornof and Anna Cavender. Eyedraw: enabling children with severe motor impairments to draw with their eyes. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 161–170, New York, NY, USA, 2005. ACM.

[56] M. Donegan, L. Oosthuizen, R. Bates, G. Daunys, J.P. Hansen, M. Joos, P. Majaranta, and I. Signorile. D3.1 User requirements report with observations of difficulties users are experiencing. Technical report, COGAIN, 2005.

[57] Renzo Andrich, Valerio Gower, Antonio Caracciolo, Giovanni Del Zanna, and Marco Di Rienzo. The DAT Project: A Smart Home Environment for People with Disabilities. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer, editors, *Computers Helping People with Special Needs*, volume 4061 of *Lecture Notes in Computer Science*, pages 492–499. Springer Berlin / Heidelberg, 2006.

[58] Tiiu Koskela and Kaisa Väänänen-Vainio-Mattila. Evolution towards smart home environments: empirical evaluation of three user interfaces. *Personal Ubiquitous Comput.*, 8(3-4):234–240, 2004.

[59] R. Bates, G. Daunys, A. Villanueva, E. Castellina, G. Hong, H. Istance, A. Gale, Lauruska, O. V. Spakov, and P. Majaranta. D2.4 A survey of Existing 'de-facto' Standards and Systems of Environmental Control. Technical report, COGAIN, 2006.

[60] R.N. Haber and M. Hershenson. *Psychology of Visual Perception.* Holt, Rinehart and Winston, 1973.

[61] Robert J. K. Jacob. Eye Tracking in Advanced Interface Design. In W. Barfield and T. Furness, editors, *Advanced Interface Design and Virtual Environments*, pages 258—288. Oxford University Press, 1995.

[62] Colin Ware and Harutune H. Mikaelian. An evaluation of an eye tracker as a

device for computer input. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pages 183–188, New York, NY, USA, 1987. ACM.

[63] M. Donegan, L. Oosthuizen, G. Daunys, H. Istance, R. Bates, I. Signorile, F. Corno, A. Garbo, L. Farinetti, E. Holmqvist, M. Buchholz, M. Joos, J.P. Hansen, D. MacKay, R. Eskillson, and P. Majaranta. D3.2 Report on features of the different systems and development needs. Technical report, COGAIN, 2006.

[64] F. Razzak, E. Castellina, and F. Corno. Environmental Control Application compliant with COGAIN Guidelines. In *COGAIN 2009 - Gaze Interaction For Those Who Want It Most*, pages 31–34, 2009.

[65] Vildan Tanriverdi and Robert J. K. Jacob. Interacting With Eye Movements In Virtual Environments. In *CHI 2000: Proceedings of the SIGCHI conference on Human factors in computing systems conference*, pages 265–272. Addison-Wesley/ACM Press, 2000.

[66] C. Abras, D. Maloney-Krichmar, and J. Preece. User-centered design. In *In Bainbridge, W. Encyclopedia of Human-Computer Interaction*, pages 1–14. Thousand Oaks: Sage Publications, 2004.

[67] R. Bates and O. Spakov. D2.3 Implementation of COGAIN Gaze Tracking Standards. Technical report, COGAIN, 2006.

[68] Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction (3rd Edition)*. Prentice Hall, 3 edition, December 2003.

[69] Adam Nathan. *Windows Presentation Foundation Unleashed (WPF) (Unleashed)*. Sams, Indianapolis, IN, USA, 2006.

[70] D. Bonino and F. Corno. DogSim: A state chart simulator for Domotic Environments. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 208–213, March 2010.

[71] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley, 2 edition, May 2008.

[72] Jan M. V. Misker, Jasper Lindenberg, and Mark A. Neerincx. Users want simple control over device selection. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, sOc-EUSAI '05, pages 129–134, New York, NY, USA, 2005. ACM.

[73] P. Huang. Promoting wearable computing: A survey and future agenda. In *Proceedings of the International Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges*, IS2000, November 2000.

[74] A. Pentland. Wearable intelligence. *Scientific American*, 9(4), 1998.

[75] Joshua R. Smith, Kenneth P. Fishkin, Bing Jiang, Alexander Mamishev, Matthai Philipose, Adam D. Rea, Sumit Roy, and Kishore Sundara-Rajan.

Rfid-based techniques for human-activity detection. *Communication of the ACM*, 48(9):39–44, September 2005.

[76] Uwe Maurer, Anthony Rowe, Asim Smailagic, and Daniel Siewiorek. Location and Activity Recognition Using eWatch: A Wearable Sensor Platform. In Yang Cai and Julio Abascal, editors, *Ambient Intelligence in Everyday Life*, volume 3864 of *Lecture Notes in Computer Science*, pages 86–102. Springer Berlin / Heidelberg, 2006.

[77] Chris Harrison, Brian Y. Lim, Aubrey Shick, and Scott 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, pages 941–944, New York, NY, USA, 2009. ACM.

[78] Jungsoo Kim, Jiasheng He, K. Lyons, and T. Starner. The gesture watch: A wireless contact-free gesture based wrist interface. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 15–22, 2007.

[79] Dong-Woo Lee, Jeong-Mook Lim, J. Sunwoo, Il-Yeon Cho, and Cheol-Hoon Lee. Actual remote control: a universal remote control using hand motions on a virtual menu. *IEEE Transactions on Consumer Electronics*, 55(3):1439–1446, August 2009.

[80] G. Bestente, M Bazzani, A. Frisiello, A. Fiume, D. Mosso, and L. M. Pernigotti. Dream : Emergency monitoring system for the elderly. In *6th International Conference of the International Society for Gerontechnology (ISG-08)*, Pisa, Italy, 06 2008.

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

[82] Berry Eggen, Gerard Hollemans, and Richard van de Sluis. Exploring and Enhancing the Home Experience. *Cognition, Technology and Work*, Volume 5, Number 1 / April, 2003:44–54, 2003.

[83] G. Demiris, M. Rantz, M. Aud, K. Marek, H. Tyrer, M. Skubic, and A. Hussam. Older adults' attitudes towards and perceptions of "smart home" technologies: a pilot study. *Medical Informatics and the Internet in Medicine*, 29(2):87–94, 2004.

[84] Dario Bonino and Fulvio Corno. Rule-based intelligence for domotic environments. *Automation in Construction*, 19(2):183 – 196, 2010.

[85] John Herbert, John O, and Xiang Chen. A context-sensitive rule-based architecture for a smart building environment. *Future Generation Communication and Networking*, 2:437–440, 2008.

[86] David Leake, Ana Maguitman, and Thomas Reichherzer. Cases, context, and comfort: Opportunities for case-based reasoning in smart homes. In Juan Augusto and Chris Nugent, editors, *Designing Smart Homes*, volume 4008 of *Lecture Notes in Computer Science*, pages 109–131. Springer Berlin / Heidelberg, 2006.

[87] U.S. Department of Energy. 2008 buildings energy data book. Technical report, Buildings Technologies Program Energy Efficiency and Renewable Energy, 2009.

[88] Wokje Abrahamse, Linda Steg, Charles Vlek, and Talib Rothengatter. A review of intervention studies aimed at household energy conservation. *Journal of Environmental Psychology*, 25(3):273 – 291, 2005.

[89] Wiluam O. Dwyer, Frank C. Leeming, Melissa K. Cobern, Bryan E. Porter, and John Mark Jackson. Critical Review of Behavioral Interventions to Preserve the Environment Research Since 1980. *Environment and Behavior*, 25/5:275–321, 1993.

[90] G. Wood and M. Newborough. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. *Energy and Buildings*, 35(8):821 – 841, 2003.

[91] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. Wonderweb deliverable d18 ontology library (final). Technical report, Laboratory For Applied Ontology - ISTC-CNR, 2003.

[92] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 2–9, New York, NY, USA, 2001. ACM.

[93] Muo, Measurement Units Ontology, Working Draft - DD April 2008.

Appendix A

Detailed Energy Survey Results

Table A.1: Results for the Direct Power Feedback questions group

Question	**Replies**	**Percentage**
1. What could be the maximum power allocation defined for the home in the video?	1.5 KW	1.71%
	6.0 KW	9.38%
	2.7 KW	32.56%
	3.0 KW	50.81%
	I don't know	5.54%
2. When was the power consumption highest?	From 6:00 a.m. to 6:05 a.m.	0.81%
	From 12:00 p.m. to 12:05 p.m.	2.62%
	From 6:00 p.m. to 6:05 p.m.	92.54%
	None of the others	4.03%
3. What appliance consumed most power?	The dishwasher	2.22%
	The washing machine	94.56%
	The fridge	2.22%
	The coffee maker	1.01%

Question	**Replies**	**Percentage**
4/5. A room is green if…	Nothing is on	71.56%
	Something is on and it has a low consumption	26.30%
	Something is on and it consumes a bit	1.22%
	What is on consumes a lot	0%
	No answer	0.92%
4/5. A room is red if…	Nothing is on	0%
	Something is on and it has a low consumption	0.30%
	Something is on and it consumes a bit	12.46%
	What is on consumes a lot	85.71%
	No answer	1.52%
4/5. A room is orange if…	Nothing is on	0.57%
	Something is on and it has a low consumption	36.68%
	Something is on and it consumes a bit	53.57%
	What is on consumes a lot	7.74%
	No answer	1.15%
6. Do the red, orange and green colors help you to understand how much you are consuming?	Yes	71.77%
	No	2.82%
	A bit	25.40%

Table A.2: Results for the Energy Goal Setting questions group

Question	**Replies**	**Percentage**
1. What is the daily energy consumption that must be respected?	1 kWh	0.71%
	3 kWh	4.74%
	5 kWh	2.12%
	7 kWh	92.44%
2. Does the actual daily energy consumption exceed the predefined limit?	Yes	94.56%
	No	4.03%
	Maybe	1.41%
3. When does the energy consumption increase?	When a new device is switched on	52.82%
	Only if there are active devices	44.46%
	When a device is switched off	2.72%
4/5. A room is green if...	I haven't consumed anything	11.01%
	Until now, the devices located in the room have consumed a little	78.29%
	Until now, the devices located in the room have consumed quite a bit	0.92%
	Until now, the devices located in the room have consumed a lot	1.53%
	The consumption meter is still green	7.03%
	No answer	1.22%

Question	Replies	Percentage
4/5. A room is red if...	I haven't consumed anything	0.90%
	Until now, the devices located in the room have consumed a little	2.11%
	Until now, the devices located in the room have consumed quite a bit	12.05%
	Until now, the devices located in the room have consumed a lot	80.42%
	The consumption meter is still red	3.61%
	No answer	0.90%
4/5. A room is orange if...	I haven't consumed anything	2.35%
	Until now, the devices located in the room have consumed a little	19.94%
	Until now, the devices located in the room have consumed quite a bit	60.70%
	Until now, the devices located in the room have consumed a lot	10.56%
	The consumption meter is still orange	4.99%
	No answer	1.47%
6. In the previous question, what do you mean for "a little"?	Less than 1 kWh	41.30%
	Less than 3 kWh	12.65%
	Less than the total energy consumption objective (7 kWh)	2.37%
	Less than the energy consumption objective associated to the room	43.28%
	No answer	0.40%

Question	Replies	Percentage
6. In the previous question, what do you mean for "a lot"?	More than 3 kWh	21.17%
	More than 5 kWh	4.40%
	More than the total energy consumption objective (7 kWh)	11.11%
	More than the energy consumption objective associated to the room	62.47%
	No answer	0.84%
7. Do you think that every room changes its color with the same energy consumption values?	Yes	18.04%
	No	68.55%
	Maybe	13.41%
8. How do rooms change color?	On the basis of the energy consumed until now	26.71%
	On the basis of the total energy consumption objective	7.76%
	On the basis of the total energy consumption objective referred to a single room	50.40%
	On the basis of devices being switched on	13.61%
	No one of the others	1.51%
9. Do the green, red and orange colors help you in understanding how you are behaving with respect to your energy goals?	Yes	43.53%
	No	13.10%
	A bit	43.55%
10. If today I've met my energy consumption goal, how shall the goal of tomorrow be defined?	Equal to today's objective	32.56%
	Lower than today's objective	65.02%
	Higher than today's objective	2.42%
11. Do you think that the next energy consumption objective shall take in account how much you exceeded the goal for today?	Yes	72.58%
	No	18.75%
	Maybe	8.67%

Question	Replies	Percentage
12. How do you like to take into account the energy consumption excess?	Decreasing the new objective with the whole today's energy excess	29.81%
	Decreasing the new objective with a part of today's energy excess	37.27%
	Increasing the new objective with the whole today's energy excess	8.57%
	Increasing the new objective with a part of today's energy excess	19.25%
	Other	5.09%
13. Do you think you shall be rewarded when your energy consumption is lower than the daily objective?	Yes	36.59%
	No	54.13%
	Maybe	9.27%
14. How do you like to be rewarded?	Increasing the new objective by the entire energy saving achieved today	7.74%
	Increasing the new objective by a part of the energy saving achieved today	16.40%
	Decreasing the new objective by the entire energy saving achieved today	17.08%
	Decreasing the new objective by a part of the energy saving achieved today	33.49%
	Other	25.28%

Appendix B

Publications

B.1 Book Chapters

1. Dario Bonino, Fulvio Corno, Luigi De Russis (2013) **Real-time Big Data Processing for Domain Experts, An Application to Smart Buildings** in: Big Data Computing, pages 33, Chapter 14, Section V, Taylor & Francis Group/CRC Press, ISBN: 978-1-46-657837-1

B.2 International Journals

1. Dario Bonino, Fulvio Corno, Luigi De Russis, Gianni Ferrero (in press) **JEERP: Energy-aware Enterprise Resource Planning** in: IT Professional, IEEE, pages 6, ISSN: 0018- 9162, DOI: 10.1109/MITP.2013.22

2. Dario Bonino, Fulvio Corno, Luigi De Russis (2012) **Home Energy Consumption Feedback: A User Survey** in: Energy and Building, Elsevier, vol. 47C, pages 11, ISSN: 0378-7788, DOI: 10.1016/j.enbuild.2011.12.017

3. Dario Bonino, Luigi De Russis (2012) **Mastering real-time big data with stream processing chains** in XRDS: Crossroads, The ACM Magazine for Students, Volume 19 Issue 1, Fall 2012, pages 4, ISSN: 1528-4972, DOI: 10.1145/2331042.2331050

4. Dario Bonino, Emiliano Castellina, Fulvio Corno, Luigi De Russis (2011) **DOGeye: Controlling your Home with Eye Interaction** in: Interacting with Computers, Elsevier, pages 15, Vol. 23/5, ISSN: 0953-5438, DOI: 10.1016/j.intcom.2011.06.002

B.3 Proceedings

1. Luigi De Russis, Dario Bonino, Fulvio Corno (2013) **The Smart Home on Your Wrist** in: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication (Ubicomp '13 Adjunct), ACM, New York, NY, USA, pages 8, DOI: 10.1145/2494091.2497319

2. Dario Bonino, Fulvio Corno, Luigi De Russis (2012) **dWatch: a Personal Wrist Watch for Smart Environments** in: Procedia Computer Science, Elsevier, 3rd International Conference on Ambient Systems, Networks and Technologies, Niagara Falls, Ontario, Canada, 27th-29th August 2012, pages 8, Vol. 10, ISSN: 1877-0509, DOI: 10.1016/j.procs.2012.06.040

3. Dario Bonino, Fulvio Corno, Luigi De Russis (2011) **A User-Friendly Interface for Rules Composition in Intelligent Environments** in: Ambient Intelligence - Software and Applications, Springer Berling (DEU), International Symposium on Ambient Intelligence, Salamanca (ES), 6th - 8th April 2011, pages 5, Vol. 92, ISBN: 9783642199363, DOI: 10.1007/978-3-642-19937-0_27