Analysis and Requirements Gathering of User Interfaces for Home Automated Systems

Akash Bhrat Manilal[1] and Paulo Carreira[2]
[1]*Department of Computer Science and Engineering, Technical University of Lisbon, Lisbon, Portugal*
[2]*INESC-ID, Lisbon, Portugal*
{*akash.manilal, paulo.carreira*}@*tecnico.ulisboa.pt*

Keywords: Building Automation Systems, Home Automation Systems, User Interfaces, Smart Environment, Human-Machine Interaction, Usability.

Abstract: As Home and Building Automation Systems become more widespread, the need for more natural forms of interaction also increases. However, many of these technologies are brittle and do not adapt to the user's explicit or implicit wishes. Indeed, variety of devices and functionalities required in a smart environment makes it difficult to integrate all controls together under a single UI, making it confusing, overwhelming and hard to operate. Moreover, there are no clear guidance when it comes to designing UI for Building and Home Automation Systems. This document analyses existing User Interfaces for Home Automation Systems to uncover their fundamental aspects of interaction. An action plan will be defined in order to develop the ideal UI. In this plan, studies of previous attempts to improve BAS Interfaces will also be presented.

1 INTRODUCTION

Automation Systems have managed to expand their features to mostly all devices and gadgets. Nowadays, turning a light on, increasing the TV volume or even heating the oven can be accomplished with a single application. As a consequence, more controls are necessary to command each desired feature. Physical interfaces are cluttered, displaying a lack of usability and adaptability. With the increasing ubiquity of interactive computer systems, usability becomes increasingly an important issue. Minor usability problems can scale to having major economic and social consequences (Mayhew, 1999). Whereas in terms of functional requirements systems have been showing evolution (Ghanam et al., 2009), we must give importance to the various aspects of interaction between the user and the application.

Interfaces for Home Automation have tried to focus on responsive behaviour in order to make them more appealing. Human-Machine Interaction is an interdisciplinary area of applied research and design practice. Its key concern is to understand the creation of User Interfaces for machines as experienced and manipulated by human users. It draws many established areas of science and craft: psychology, computer science, anthropology, industrial design (Carroll, 1991).

This document will address the most frequent problems related to Home Automation Systems (HAS) Users Interface, such as Hysteresis, Manual Override, notion of Progress, Completion and much more. In the end, a design proposal will be provided. A description of the procedure that will be used in the creation of a new User Interface (UI) focused on HAS as well as the evaluation method to confirm it's effectiveness.

1.1 Goals

The goal of this paper is to resolve some existing problems related to usability, user friendliness and feature management of Home Automated Systems and build the most complete User Interface.

To implement our idea we have to identify functional requirements (all the features a house has to offer), non-functional requirements (Usability, Accessibility, Portability, etc.) , all the possible modalities (all the devices it can be implemented) and the UI patterns we can apply.

For house features purposes we will split into the following categories: White Goods (everything that has to do with electronic Equipment), Illumination, HVAC, Environmental (analysing external factors that can influence, for example, illumination inside the house) and Security. One of the problems we will face is how to build a single User Interface for all those categories and still make it user friendly.

1.2 Problem Definition

Home automation and controls provide advanced functionality to homes using distributed control systems. Demand for energy efficient solutions, enhanced security, increased venture capital funding, and greater convenience are some of the factors that have led to the growth of the home automation and controls market. The market is, currently, at the high growth stage of the industry life cycle, and is expected to remain in this phase till 2018.

The majority of existing Home and Building Automation Systems focus on functionality gathering, house features, integrating with external systems or even improving response time. Even though they are significant points, they conceal other requirements that also are of major importance to the proper functioning of systems and acceptance from users.

With this work we propose to resolve the following issues: notion of progress and completion (real-time feedback), Hysteresis and Manual Override. Giving the user a accurate progression state during the course of a requested action allows him to have a real-time knowledge of the state of his request. The same goes to the notion of completion. Solving this problem has a great impact on Home and Building Automation Systems (HBAS) since they can make use of this solution to any kind of request.

Hysterisis is the dependence of a system not only on it's current environment but also on it's past one. Regarding BHAS, we can identify Hysteresis problems when a user wants to perform an action that contradicts a previous one. Solving this problem, we would be not only conserving the system itself but also the physical hardware that could be affected by it.

Manual Override is an action where the control is taken from an automated system and given to the user. The conflict resides in what the system should do, either gives full responsibility to the user, allowing him to fully override any scheduled event or refuses to cede control, making the user less empowered.

Although there are several other issues to be identified, we believe that solving at least these three would show great improvements for Home Automation Systems.

2 CONCEPTS

User Interface is the visual part of computer application through which a user interacts with a software. It determines how commands are given to the computer or the program and how information is displayed on the screen. However, the field that studies how UI's should be designed is much wider than it's definition. In this section we will introduce the concept of Smart homes and Human-Machine Interaction. Both of them will be related with UI designs referencing Ten Usability Heuristics.

2.1 Smart Homes

Smart homes go by several names, including Integrated Control Systems (ICS) and Home Automation Systems (HAS). By any name, these systems are used to control devices around the home. Home electronics and appliances including doors, lights, surveillance systems and consumer electronics are some examples (Humphries et al., 1997). Control Systems can also provide information, meaning, users can find out how much electricity they have used on specific appliances and utilities can read meters remotely (Han et al., 2010). An important goal of smart home research becomes how to appropriately expand system capabilities to produce more control both perceived and actual (Davidoff et al., 2006). To provide mobility, usually, users are able to control with a smart-phone or tablet. However, these systems lack user friendliness and are neither intuitive nor realistic (Wimsatt et al., 2006).

Domestic technologies have been adopted at different rates throughout the years. To better understand the direction smart home technologies are taking and what programmers and scientist believe is the next evolution for HAS (Harper, 2003), it is important that we distinguish the concepts of *time-saving* goods and *time-using* goods (Bowden and Offer, 1994). The first ones are meant to reduce the time needed to complete a task, in order to have more time for yourself. Vacuum cleaners, washing and drying machines are some examples. The majority of these goods took decades to diffuse and all of them are clearly related to household income. As for time-using goods are those which occupy free time and improve its quality by giving the user more to entertain. Television, radio and video are the most common and well-known time-using goods for most people. Unlike time-saving goods, these have spread quickly and showed much less relationship to household income.

The relation between these two goods rely on the time they occupy. The more people spend time using goods like television or radio, the less they have to complete house tasks. This can be considered the motivation needed to start thinking about smart homes.

2.2 Human-Machine Interaction

Human-Machine Interaction (HMI) is described as the interaction between human users and machines. The abbreviation HMI should be considerably extended to cover the ergonomics challenges presented by the contemporary, highly complex and dynamic human-machine systems. The notion remains a key point with its properties of friendliness, usability, transparency, etc. However, cognitive ergonomics has developed a large number of studies on human automation relationship failures, which concern what is behind the interface (Baecker and Buxton, 1987).

The degree of automation in control of dynamic systems has substantially been increased over the last decades. This is true for all technical systems. High levels of safety, performance and efficiency have been achieved by means of the raise use of automatic control. The need for improved Human-Machine Interaction (HMI) spread proportionally to the degree of automation. Generally, increased automation does not replace human users who are interacting with the machine, but shifts the location of the interface between both. Higher complexity and more sophisticated control structures require a new quality of communication and co-operation between human and machine. From the front line humans operator point of view, machines are not only tools but also possibly autonomous agents and some coherence must be maintained between humans and machines actions on the environment, regardless the interface (Johannsen,).

2.2.1 Usability Heuristics

We can analyse UI's through analysis techniques, computerized procedures, empirical methods and *heuristic methods* (Nielsen, 1994; Nielsen and Molich, 1990). The last one consists in presenting an UI to a number of people that are asked to evaluate a set of usability principles. The original set of heuristics used for several early studies was developed with the main goal of making the method easy to teach (Nielsen and Molich, 1989). In order to normalize the evaluation set of usability principles, Jakob Nielsen defined what is considered to be the Bible of Usability Evaluation:

Visibility of System Status, meaning, the system should always keep users informed about what is going on, through appropriate feedback within reasonable time;

Real-world Conventions should be followed, making information appear in a natural and logical order (words and phrases);

Supporting Undo and Redo is necessary in case users choose system functions by mistake and need to leave the unwanted state without having to go through an extended dialogue;

Consistency and Standards prevent users from wondering whether different words, situations, or actions mean the same thing;

Eliminating Error-prone conditions or showing confirmation boxes before users take an action will help make it clearer;

Minimize the user's Memory Load by making objects, actions, and options visible. Instructions for use of the system should be visible or easily retrievable whenever appropriate;

Flexibility and Efficiency of Use for more experienced users, the system should allow them to speed up the interaction so they can execute the desired task;

Stick with what Really Matters . Dialogue boxes should not contain irrelevant information. Overflow of information might confuse the user and will eventually hide the focus point;

Help users Recognize, Diagnose, and Recover from Errors. Error messages should be expressed in plain language, precisely indicating the problem, and constructively suggesting a solution;

Provide Help and Documentation even though it is better if the system can be used without it. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

2.3 Fundamental Aspects of Interaction with HAS

2.3.1 Feedback Issues

Feedback is the process part of a chain cause-and-effect. Wondering what a software program is doing is a common way to confirm that UI lacks clear information. Feedback principles in a UI needs to be immediate and synchronized with user action. As for Home and Building Automation Systems (HBAS) there are a specific set of responses an UI should give to their users.

Hysterisis is the dependence of a system not only on it's current environment but also on it's past one. This dependence is due to conflicts on internal states.

To avoid conflicts, it's history must be known (Visintin, 1994). We can identify this problem every time a user requests a functionality that is exactly the opposite of the previous one in a short amount of time. Opening and closing blinds and turning the heater on and off are some examples of such problem. To an HBAS it is relevant to determine whether the application should not allow the user to complete his task or simply showing a warning message to acknowledge the user that his request might cause conflicts.

Other kinds of feedback issues come from Progress and Completion of tasks. For the first one we can show the user an accurate progression of the requested action by measuring the average time to complete the activity (Best Effort). Once the average time has passed, a message is shown to the user, confirming the completion of the task. This approach may lead the user in error since we cannot be certain the task is already completed. The second option is to give the Feedback Bus the responsibility of sending and acknowledge (ACK) signal to confirm when the task is completed. The last choice and the most common one, is using a motion sensor in order to get physical feedback. This approach can cause issues if an object is preventing the motion sensor to read correctly.

Manual Override is an action where the control is taken from an automated system and given to the user. When building HBAS, the main goal is to give the user as less work as possible to perform tasks. However, the system should be prepared to deal with exceptions. On extreme cases, giving the machine full responsibility for automations of all features of the house can become an issue. If they follow rules installed by the manufacturer or required by law and refuse to cede control in some situations then the owners of the devices may feel less empowered, alienated and lacking true ownership.

The last two feedback issues are related to Conditions and Energetic Behaviour. The first one is an issue frequently related to temperature. Considering a scenario where an user is using an application connected to all devices in the house, including HVAC, and pretends to turn the heater on. As the temperature rises, the system should be able to control it so it doesn't get uncomfortable to the user (too hot or too cold). Energetic Behaviour relates to how the user deals with his house. The UI should be capable of changing users behaviour in order to save electricity and water.

2.3.2 Control Abstractions

There are many ways users can interact with the system. Each of them can have different results and raise distinct issues. Control abstractions can be distinguish between Spatial, Equipment and Services. For each one of them a different UI should be presented since they possess distinct features.

Spacial requests are what we call Scenarios. They allow the user to change multiple features at the same time in order to create the environment he wishes. Taking an everyday situation such as a user wanting to watch a film, there should be a scenario that is able to dim the living room lights, turns on the tv, turns off any other light and starts the film. Spacial requests involves the whole house and all its features. To represent this control in an UI, besides showing the possible scenarios, the layout should contain the details of each of them.

Controlling Equipments allows the user to make system requests targeted to a specific equipment. The action to be applied to the equipment does not have to be detailed. Requests like dimming all the lights to twenty percent are considered equipment requests.

Requesting for a service is the same as saying that the user wants a specific task to be performed. For example, turning on the lights. These types of services are what make the UI vast and showing too many of these can make the UI less User Friendly and very confusing. It is important that we organise all features with categories so the user finds it easier to request a specific service.

3 RELATED WORK

In this section we will address theoretical studies related to HAS characteristics and their User Interfaces. Firstly we will present the notion of portable systems and remote access. Then we will show some systems capable of controlling various house features with only one UI and in the same section we will discuss customisable UI's.

3.1 Portability and Remote Access

Regarding HAS in relation to portability, typical user interfaces for controlling home systems include computer displays with touch screens, dedicated display devices such as LCD panels with buttons or a touch screen or hand-held remote controllers. All of these examples are devices which are dedicated for control only. The users require extra space and incur extra expenses for them. In some cases, these display devices are installed into walls. In these cases, special installation work is necessary and it becomes very difficult to modify the devices and/or their location. In the case of hand-held remote controllers, the display capabilities are limited and the functions are typically dedicated to specific purposes such as lighting control. Portability began to matter and attempts where made in order to overcome those drawbacks.

In 1996 a method for menu-driven UI, more particularly, for distributing menus throughout a home, was presented (Fujita and Lam, 1996). In order to make the software portable, functions could be implemented using a personal computers, infra-red, remote controls, home distributed networks and television receivers. This invention was advantageous in a way that menus could be distributed to any place in a home, users could control the home system and outside services from any display location in the home and different menus could be distributed at the same time to distinct display locations.

In 2005, a mobile-based HAS solution was introduced (Van Der Werff et al., 2005). The design consisted on a mobile phone with Java applications, a cellular modem, and a micro-controller. User friendly graphical UI was provided on the mobile phone through applications developed in Java programming language. The controller board resided at home and worked as a home server, which carried out the task of operating and monitoring house appliances. The home server communicated with the remote control via cellular modem.

More recently, a ZigBee-Based Home Automation System was proposed (Gill et al., 2009). ZigBee is a specification for a set of communication protocols used to build personal area networks built from series of small, low powered digital radios (Alliance, 2006). It is based on an IEEE 802.15 standard. ZigBee devices often transmit data over longer distances by passing data through intermediate devices to reach more distant ones, creating a network with no centralized control or high-power transmitter/receiver able to reach all of the networked devices. The decentralized nature of such wireless *ad hoc* networks make them suitable for applications where a central node can't be relied upon.

Another key feature for HAS is the capability for *remote operation*. Considerable efforts have been put into the development of remote control systems for home automation. With the proliferation of Internet, various Internet-based remote control architectures for home automation have been proposed (Al-Ali and Al-Rousan, 2004; Corcoran and Desbonnet, 1997). These systems rely on the Internet as the medium for communication and generally feature friendly graphical user interfaces. Home gateways provide network interoperability, a simple and flexible user interface, and remote access to the system.

3.2 All in One and UI Customisation

Home Automated Systems have been making an effort to control all house features. The standard approach now commonly used is for each device or system in a given environment to be controlled according to a particular methodology which might differ dramatically from other systems (Stein et al., 2000). For example, a home might include a security system, an entertainment system, an environmental control system, and so forth, each with it's own unique interface. By requiring the user to learn several methods of operating each system or set of devices in the environment, it is more difficult for the user to become familiar with the various systems and to take full advantage of all their features. Another drawback associated with this standard approach is that the use of different interfaces may result in an increase in the amount of space taken up in the setting. More and more users want a single application capable of controlling Illumination, HVAC and white goods.

Liu and Xian approached this problem particularly for disabled people (Liu and Xian, 2007). The system was build so as to merge all house features. To enhance the experience, a voice controlled, User Friendly UI was carefully designed, so that user could use not only standard keyboard and mouse, but also the voice to control home environments including lights, TV, radio, VCD/DVD player, fan, and air conditioner. Furthermore, the self-designed software applied home automation control idea to internet access and PC application software access with the features of surfing the internet, sending and receiving emails, using other PC software such as Microsoft Office.

A report by William Wimsatt went further and in addition to gathering features, developed a contextual UI (Wimsatt et al., 2006). Contextual Interface is a product of a design process in which the various interface features are selected to improve the users ability to operate the interface. Examples of interface features that can be controlled include the selection of controls, size, shape and position, background and foreground colour schemes, sounds, as well as the selection of actions that are initiated by operating a control. In most graphical user interfaces these items remain static so that irrespective of the context, a control such as a *start* button, remains in the same place on a display with the same appearance and performs the same function. This invention involved two levels of contextual relevance.

First, the user interface as a whole is contextually sensitive to the elements appearance (e.g., colour, size, font, etc.) and their behaviour vary depending on the context of the control unit (CU). The context of

Table 1: Systems Comparison. Each column represents the dimensions analysed while the lines represent the systems/works mentioned on the previous sections.

	Portability	Remote Access	All-in-One	Custom UI
Menu Driven UI	+	--	+/-	--
Mobile-based HAS	+	+/-	+	--
Zig-Bee	+	+	+	--
HAS for Disabled	+	--	+	--
HAS Contextual UI	--	+	+	+

the CU is represented by state information known to it, including context-specific and global information known to a single or multiple CU's in a system.

And second, individual elements are themselves made contextually sensitive. In a particular implementation, contextual sensitive interface elements include interactive screen elements such that a single screen element can simultaneously display information about the context (e.g., current temperature, sound volume, light level, etc.) as well as implementing behaviour to send messages to a controlled system that can affect the displayed information (e.g., a thermostat, an entertainment system, or a lighting subsystem).

3.3 Discussion

In order to identify the common issues and strengths of each work described during this section, a comparison was made. Table 1 shows the results of the analysis based on the details of each work. To facilitate the comparison, a convention of symbols with different meanings were defined:

"+" the system fully supports the feature to be evaluated;

"+/-" the system partially supports the evaluated feature;

"–" either the system does not support the feature that is being tested or the work does not present any information regarding the evaluated dimension.

Examining the table we can see that each dimension has different results. Regarding portability, we can verify that all of them, excluding the work of William Wimsatt (Wimsatt et al., 2006) (which doesn't present any information related to this dimension), are portable. The systems were built in a way that allows users to access it in different devices. Some of them use infra-red, remote controls and home distributed networks to accomplish this feature. In contrast, almost none of them are customisable.

UI Customisation is the common issue among most systems. When building User Interfaces, systems tend to create fixed layouts that they believe to be ideal, usability wise. To surpass this problem, Contextual User Interfaces was used (Wimsatt et al., 2006). Contextual Interface is a product of a design process in which the various interface features are selected to improve the users ability to operate the interface. Examples of interface features that can be controlled include the selection of controls, size, shape and position, background and foreground colour schemes, sounds, as well as the selection of actions that are initiated by operating a control.

Forcing a user to have a system that controls irrigation, another that controls water and another for illumination nowadays is not viable. Companies understood this problem and made an effort to join all those systems into one, requiring the user to adapt a single UI. According to table 1 we can see that all of them took this problem into account and gather all the functionalities.

The last analysed feature was the remote access. In the last few years, companies have been trying to make their software as accessible as possible, using the network as an asset. According to the table, results vary from work to work.

4 EVALUATION OF EXISTING SYSTEMS

HAS has been growing exponentially in the last years. The proof relies in the various Automation Systems available in the market. The last few years there's been an increase of requests to implement Integrated Control Systems (or Smart homes) and at the same time, an increase of companies capable of providing and supporting those systems. In order to evaluate existing systems we've decided to choose six HAS that stood out among the others: mControl, HomeSeer, Control4, PowerHome2, Vivint and ActiveHomePro.

4.1 Evaluation Method

In order to fairly evaluate the systems we there was a need to define a set of points (or dimensions) of evaluation, each of them containing a set of specific features. For every dimension of evaluation, a system can get minimum score of 0 or a maximum of 5. We decided to also include an "Observations" criteria, that allows the system to gain or loose 0.5 points. If we find features that we did not take into account in our evaluation but are worth mentioning, whether it's good, bad or some detail, it will be considered.

4.1.1 System Features

This evaluation point will take into account house features that the system allows you to control. For evaluation purposes, we consider that are four essential features:

Lighting and illumination of the whole house. The system should also give the possibility to control brightness;

Heating, Ventilation and Air Conditioning(HVAC) The software should be able to control home's heating and cooling system as well as give information about current state;

Motion Sensors allows the system to detect movement or monitor occupancy status throughout the house. This feature, combined with any other, can be very helpful to support other functionalities;

White Goods or whiteware, relates to heavy consumer durables such as air conditioners, refrigerators, stoves, etc., which used to be painted only in white. Despite their availability in varied colours now, they are still called white goods;

4.1.2 User adaptability

The UI has to be clever enough to adapt users demands. In this point, for the software to get the maximum score it has to able to schedule events (timed events) so the user has the ability to prepare everything with no worries. It has to have scenarios. A scenario is a feature that allows the user to control the space around him (Spatial Control Abstraction). They trigger multiple events at the same time with a single button.

The last two points of evaluation are UI Customisation and Session-based application. The first one concerns the UI's ability to be modified. Letting the user customise the main panel, enabling a favourites section/tab are two features to be taken into account. As for the second one, it is related to the capacity of the system to distinguish between different users inside the same house. If each user has it's own username and password the system would be able to collect different types of data, identifying the person that has spent most water and/or energy in the house or even be able to give suggestions based on what it has learned.

4.1.3 Ease of Use, Help and Documentation

Concerning the Ease of Use of an application, we rated each product on its ability to produce an intuitive user interface based on its Commands Organization. If the application uses tabs, buttons with intuitive icons, colour scheme to facilitate the differentiation between functionalities and so on. Overcomplicated processes and clunky interfaces caused systems to rank lower in this category. Another point is the Remote Access. Nowadays it is crucial that the user is able to access the application from anywhere in the world. The most usual devices that should give access are desktop computers, laptops and smartphones. The last point to be evaluated is Voice Recognition. The user shouldn't be obligated to carry his device around the house. Once he is inside he wants to communicate with it in order to satisfy his/her needs. Voice recognition allows that to happen and shows to be an extremely valuable point for this evaluation.

As for Help and Documentation, we ranked each system on the customer service options that it provided, including tutorials, user manuals, online documentation and phone support. Less common features like a live chat option and user forums can also be included.

4.2 Systems Evaluation

4.2.1 mControl v3

mControl was build thinking of all the devices a house can possibly have. This product can control Lighting, HVAC, security systems, IP cameras, irrigation systems, A/V systems, window shades and much more.

This software allows the users to choose from preset scenarios that can trigger multiple functionalities at the same time. Some of them can be automatically initiated if certain conditions are met (Sunset/sunrise events or even timers). Alternatively, you can program your own scenarios.

There isn't, however, any way to customise the UI itself, meaning, buttons layout, main menu preferences. Another flaw of mControl is that it cannot be session oriented (login/logout) and that does not allow us to know for sure who is spending what and at what rate. For example: if the software alerts the user that this week there was an increase of 30 percent on electricity there is no way to know who is accountable for that, making it hard to control or reduce that percentage.

The interface is clean and intuitive: they mostly use tabs to organise commands and abilities. There are already a pre-set role of scenarios that the user can use. These scenarios can be activated through voice recognition functions. The remote access is very useful and a big advantage because it allows the user to access the local panel by using a pc, laptop, a touchscreen device with internet connection, smartphones or browsers.

mControl is sold in modules. There is a mandatory Base Module that can be extended with Energy Management, Media Control and Professional Modules. Besides providing mobile application to access this software, mControl also is iTunes and Windows Media Center compatible. That means that you can program to turn on your tv or make it part of a scenario.

Help and Documentation is the Achilles tendon of this system. They do not offer interactive tutorials, nor phone support. Besides the user manual, mControl offers email support, online forums and FAQs and live chat option with customer service.

4.2.2 HomeSeer

HomeSeer includes all the features we were looking for and many more. It has a set of scenarios that you can customise and trigger either manually or automatically. You can even include the motion sensors in your scenarios. For example, when you pull your car out of the garage, all the lights turn off and the alarm system is activated.

Just like mControl, HomeSeer also does not allow the user to customise the UI and is not session-based. This system is known for their emphasis on ease of use. In this case, the software separates different functionalities using tabs and colour schemes, making it easier to find features. It also has voice recognition commands, as long as you implement a specific microphone throughout the house and supports remote operations.

The HomeSeer system is compatible with an enormous variety of popular and niche programs. It supports all of the major media management applications, like iTunes, Windows Media Player and Windows Media Center.

As for Help and Documentation, it offers everything we analysed and more. They have a tutorial that you can access easily through their website and a FAQ section. They also give phone support.

4.2.3 Control 4

Like the previous systems, it meets all the requirements of house features we were looking for. It has Audio and Visual control systems to complement. The big contrast between Control 4 and the previous systems is the lack of compatibility with other programs.

Control4 allows to schedule events and create custom scenarios with complex triggers (involving various functionalities from different tabs). Drawbacks: lacks compatibility with other programs, it is not customisable (in terms of UI) and is not session-based.

This system is the first one of our analyses that does not meet all requirements we established for Ease of Use. The command organization is made mostly through colour schemes and their remote access allows you to control the system any part of the world. However, it does not support voice recognition.

Help and Documentation is very poor. It only has User Manual and a FAQ section on their website to help customers in trouble. Does not have telephone, email or live chat support, making this system the worst between all the analysed ones for this dimension.

4.2.4 Power Home

The first one that does not meet all the basic requirements of our system features analysis, PowerHome is still able to cover the following house features: Lighting, Motion Sensors and most of the White Goods.

PowerHome allows the basic of scheduling events and has a pre-set number of scenarios that cover most of the user needs. It also does not allow the user to customize his UI and is not session-based.

The UI it seems to be confusing. They do not separate clearly all the features they provide nor use any kind of understandable scheme to ease the use of the application. It is not User Friendly, it is actually overwhelming and cluttered. The system offers limited support for external modules, and it doesn't have mobile applications. It offers a comparable user experience to HomeSeer, providing a clean and easily understandable interface.

When it comes to help and documentation they do not have tutorials or phone support, however they have a live chat option, user manual and a forum that can respond to almost every doubt you might have.

4.2.5 Vivint

The most impressive part of Vivints home automation system is its integration with their security systems. The Go!Control panel that comes with every Vivint system serves as the central hub for all activity, including announcements of time-based triggers and device activations. You are given more control over your devices through their online control panel. In the web-based application, you can create and modify time-based and scripted triggers for all of your devices.

Vivint is the only one of our analysis that offers something similar to session-based software. As mentioned before, the point of working with a login/logout system is for the system to acknowledge who did what and provide ways to control the consumption and every kind of expenses of the house. It allows each user to have distinct codes when using the software. With this we have a way to know who spend what and when. The disadvantages of the product are the inability to offer Software Integration and not allowing UI Customization.

It has a very good command organization, using tabs and explicit icons to clearly distinguish between all the features it can provide. The system also provides mobile applications on several platforms that you can use to control your devices from any location with an internet connection, though you cannot create or modify triggers or events from the mobile application. Vivint does not offer voice recognition commands.

In Help and Documentation point, this software is above average. Although it doesn't offer tutorials, besides user manual, FAQ section and phone support, it allows you to clear your doubts or issues through email (email support).

4.2.6 ActiveHome Pro

This system is the product of the home automation manufacturer X10s. Despite the extensive know-how backing the product it presents great lack of compatibility with other programs and devices. When it comes to house features it only includes Lighting and White Goods. There are no motion sensors or HVAC control, making the triggers and scenarios very limited.

ActiveHome Pro includes timed triggers and generic event triggers, like sunrise/sunset events. It is also possible to use events to create scenarios, as with other products. The system does not offer voice recognition or remote access. This lack of remote access is one of the more confusing things about ActiveHome Pro. All scenarios, triggers, programming and management must take place at the console or computer that you set it up on. This drawback significantly damages the usability of the product, limiting to a very narrow set of potential uses.

As for Help and Documentation, it brings every option to the table from email support to a FAQs section to user forums. They offer a vast selection of customer support options.

4.3 Results and Discussion

After the previous analysis it is clear that most of the systems include, either in modules or as a whole package, all the house features we were looking for. We need to take this information into account since our solution will be based on what most system possess.

Table 2: Overall systems comparison. Horizontally, an individual score is presented, according to each dimension of evaluation. the last cell of each line represent the total score of each system. Vertically, the last cell represent the sum of all the points acquired in a specific dimension. Darker cell represents the best dimension and system. Grey cell represent the system and dimension with the lowest total score.

	Points of Evaluation				
Products	System Features	User Adapatability	Ease of Use	Help and Documentation	
mControl	5	3,5	4,5	3,5	**16,5**
HomeSeer	5	3,5	5	5	**18,5**
Control 4	5	3	4	2	**14**
PowerHome	4	2,5	2	4	**12,5**
Vivint	5	3,5	4	4	**16,5**
ActivateHome Pro	2	2	1,5	5	**10,5**
	26	**18**	**21**	**23,5**	

According to table 2, the low scores presented on user adaptability are due to the lack of flexibility regarding UI customization. None of the presented systems offers that option to the customer. Users should be able to customise as they see fit, in order to get to the functionalities they use the most the fastest way. The other reason making this point with an average score is the fact of not being session-based applications. Most of the systems allow the user to access the software remotely. However not all allow voice commands. Tabs seem to be the preferred way to organise functionalities.

Finally, the last point analysed, help and support, showed to be a bit controversial. There are some systems with low scores on system features, user adaptability and ease of use but offer great ways to solve problems. The less the user uses help and documentation, the better he knows how to use the software. This doesn't mean we can ignore documentation or warning messages. We have to let the user know what he's doing in order to avoid making him search for the answers.

5 SOLUTION PROPOSAL

5.1 Action Steps

The research of this document, specially with respect to the conclusions drawn from our analysis of the related work, resulted in a set of requirements and assumptions in which the work is based. Some will still call for the opinion of our target users.

We know that the application is for domestic use, meaning, different age groups will interacting with it. Therefore, the UI should be User Friendly, look familiar to the user and easy to understand. The goal is to respect all the usability heuristics defined by Jakob Nielsen.

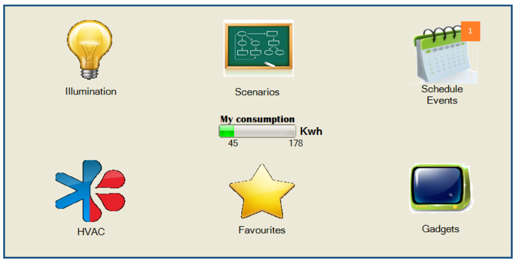

Figure 1: Main panel mockup. From top left: Illumination, Scenarios and Scheduled Events, HVAC, Favourites, Gadgets. In the middle of the screen there is a energy consumption bar where the user can see the amount of power watts spent in the house and the amount he contributed.

Another aspect we have been pointing out is the need to differentiate each user within its own house. This means that, whatever other functional requirements it might have, it has to be **session-based**. Taking into account previous research, this can be achieve with simply making the users register in the system in order to be distinguished.

Other **functional Requirements** will be subject of evaluation: Features like illumination, HVAC and White Goods. The application should also allow users to choose from presets, edit and/or create scenarios. Scheduling events it is also important. Besides that, the application should be able to verify spacial occupation through motion sensors (either inside or outside the house) so it can help regulate Illumination or simply for security reasons.

5.2 User Interface Mockups

To study our solution proposal, we have developed mockups that sketch the final interface. Figures 1, 2 and 3 present three of them.

The first one (figure 1) shows the UI main panel, containing 6 buttons: Illumination, HVAC, Scenarios, Favourites, Schedule Events and Gadgets. In the middle there is what we call a consumption bar where the user can see how much electricity was spent in general (in this case 178Kwh) and, at the same time, how much that specific user spent (45Kwh). If the user scheduled any events, the main panel will present a numeric notification referring to the number of events scheduled that are due to start.

As for figure 2, shows the Illumination tab. This layout can be seen every time a user chooses the Illumination button. On the left side, the UI will present the number of available lights in the compartment the user is (it will be possible thanks to the motion sensors and occupancy verification), meaning, this UI will change according to the room you are in. Besides that, the application will show a status of all lights in

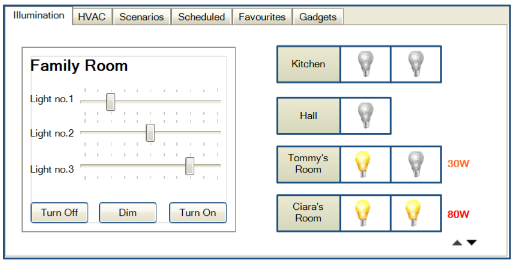

Figure 2: Illumination mockup. On top all the tabs representing main menu buttons. The left side shows all the Family Room lamps and their brightness. Below the three buttons, turn off, dim and turn on, are actions that can be taken only for that compartment. The right side shows all the other house divisions and the intensity of illumination (if any).

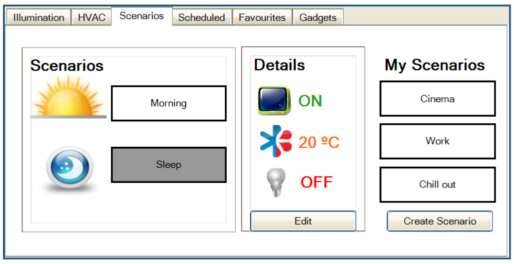

Figure 3: Scenario mockup. On top all the tabs representing main menu buttons. On the left side shows two default scenarios: morning and sleep (selected). In the middle are the specifications of the selected scenario. The right side shows all the custom scenarios created by the user.

the house, whether they are switched on or off and the total power watts of all the lamps.

Finally, 3 buttons will be available for quick action:

1. Turn Off - Turns off all the lights in the compartment the user is.
2. Dim - Dims the lights in order to adjust the brightness.
3. Turn On - Turns all the lights of the current compartment on.

At last, figure 3 represents the layout for the Scenarios button. The UI is clearly divided into three parts: the default scenarios, i.e., the ones already defined by the application itself and present no matter who's the user, details, representing the details of the chosen scenario. In this case we can see that the sleep scenario will turn off all gadgets, turn the Air conditioner to 20 degrees and turn off all the lights. Finally, the my Scenarios part shows the custom scenarios. This part may vary depending on what the user set up. It also allows the user to edit any scenario or create more.

5.3 Evaluation

To validate our solution proposal we have decided two ways to evaluate it: Target users feedback and evaluation metrics.

Target users can be divided in two types: the ones that already had experiences with other interfaces for Home Automated Systems and the ones that never had any kind of experience with this type of systems. The first ones are able to give feedback, comparing it with other systems. Not only they can point out design flaws, but they can also give advice based on ideas from other HAS. As for users that never used interfaces for HAS before, they can give a different perspective, usability wise. Since they do not have any terms for comparison, reviews are going to concentrate on the level of satisfaction and comfort our User Interface offers.

In the previous section of this document, an analysis of existing systems was made. The evaluation was based on the following dimensions: System features (Illumination, HVAC,etc.), user adaptability, ease of use and help and documentation. We are going to use these dimensions and compare the same systems that went through this evaluation.

With this two validations we are capable of validating our solution proposal not only with target users but also with systems that are well rated in the Home Automation market.

6 CONCLUSIONS

Feedback issues regarding User Interfaces for Home Automation Systems were presented. This paper aims to solve some of them, mainly: Hysteresis, Manual Override and Notion of progress and completion.

So far we have studied previous attempts to solve this problem, understood the dimension of it and conducted a study with existing HAS in the market and analysed the results regarding a set of evaluation methods defined by us. Furthermore we designed a set of mockups capable of simulating some of the feedback problems we aim to solve.

However, the proposed approach is yet to be evaluated in real life settings. We believe that the current mockups can be enhanced by a number of ways. For future work we intend to test the designed mockups in order to get users feedback.

We think that our effort in this project is a cornerstone that can inspire the building of more sophisticated and usable UI's for HAS in the near future.

REFERENCES

Al-Ali, A.-R. and Al-Rousan, M. (2004). Java-based home automation system. *Consumer Electronics, IEEE Transactions on*, 50(2):498–504.

Alliance, Z. (2006). Zigbee specification. *Document 053474r06, Version*, 1.

Baecker, R. M. and Buxton, W. (1987). Human-computer interaction. *A multidisciplinary up*.

Bevana, N., Kirakowskib, J., and Maissela, J. (1991). What is usability? In *Proceedings of the 4th International Conference on HCI*.

Bowden, S. and Offer, A. (1994). Household appliances and the use of time: the united states and britain since the 1920s1. *The Economic History Review*, 47(4):725–748.

Brown, P. J. (2003). *Software portability*. John Wiley and Sons Ltd.

Carroll, J. M. (1991). *Designing interaction: Psychology at the human-computer interface*, volume 4. CUP Archive.

Corcoran, P. M. and Desbonnet, J. (1997). Browser-style interfaces to a home automation network. *Consumer Electronics, IEEE Transactions on*, 43(4):1063–1069.

Davidoff, S., Lee, M. K., Yiu, C., Zimmerman, J., and Dey, A. K. (2006). Principles of smart home control. In *UbiComp 2006: Ubiquitous Computing*, pages 19–34. Springer.

Fujita, Y. and Lam, S. P. (1996). Distribution system and method for menu-driven user interface. US Patent 5,500,794.

Ghanam, Y., Shouman, M., Greenberg, S., and Maurer, F. (2009). Object-specific interfaces in smart homes. Technical report, Citeseer.

Gill, K., Yang, S.-H., Yao, F., and Lu, X. (2009). A zigbee-based home automation system. *Consumer Electronics, IEEE Transactions on*, 55(2):422–430.

Han, J., Yun, J., Jang, J., and Park, K.-R. (2010). User-friendly home automation based on 3d virtual world. *Consumer Electronics, IEEE Transactions on*, 56(3):1843–1847.

Harper, R. (2003). *Inside the smart home*. Springer.

Humphries, L. S., Rasmussen, G., Voita, D. L., and Pritchett, J. D. (1997). Home automation system. US Patent 5,621,662.

Johannsen, G. Human-machine interaction.

Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys (CSUR)*, 24(2):131–183.

Liu, D. and Xian, D. (2007). Home environmental control system for the disabled. In *Proceedings of the 1st international convention on Rehabilitation engineering & assistive technology: in conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting*, pages 164–168. ACM.

Mayhew, D. J. (1999). The usability engineering lifecycle. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '99, pages 147–148, New York, NY, USA. ACM.

Mooney, J. D. (1995). Portability and reusability: common issues and differences. In *Proceedings of the 1995 ACM 23rd annual conference on Computer science*, pages 150–156. ACM.

Nielsen, J. (1994). Heuristic evaluation. *Usability inspection methods*, 17:25–62.

Nielsen, J. and Molich, R. (1989). Teaching user interface design based on usability engineering. *ACM SIGCHI Bulletin*, 21(1):45–48.

Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM.

Prieto-Díaz, R. (1993). Status report: Software reusability. *Software, IEEE*, 10(3):61–66.

Spencer, D. (2004). What is usability? *University of Melbourne, Melbourne*, pages 108–115.

Stein, M., Kaufman, T. R., Richarz, Y. A., Tarlow, K. A., and Nesbitt, B. C. (2000). User interface for home automation system. US Patent 6,140,987.

Van Der Werff, M., Gui, X., and Xu, W. (2005). A mobile-based home automation system. In *Mobile Technology, Applications and Systems, 2005 2nd International Conference on*, pages 5–pp. IEEE.

Visintin, A. (1994). *Differential models of hysteresis*. Springer Berlin.

Wimsatt, W. et al. (2006). Home automation contextual user interface. US Patent 7,047,092.