How Smart Are Our Homes?

Andres Albanese and Celina S. Albaneses [*]

1801 University Ave 407, Berkeley, CA 94703, USA
andres.celina@mac.com
http://www.panare.com

Abstract. This work measures home automation implementations in terms of Comfort, Smartness, and Performance. These measures are easily computed from counting the loads, buttons, and scenarios present in home automation implementations. The works analyzes two novel automation programs to clarify these measures and to compare home automation implementations. We describe a novel use of available home controllers to implement scenarios triggered by event sequences. We provide examples of real deployment of home controller programs that can be used in residential and commercial applications.

Key words: Home Networks, Residential Gateways, Smart Homes, Hidden Commands, Event Sequence Traps, Home Controllers

1 Introduction

Home automation (HA) implementations are diverse and difficult to compare because they use different technologies. The electrical wiring of homes depends on the town wiring code, electrical standards and construction cost. A simple house has loads and basic switches to control the loads. Loads are defined as lights or group of lights in a common circuit, motors, and appliances.

Fig.1 shows an automated house implemented with My Home component [1]. This implementation has addressable components interconnected by an alarm bus, and an automation bus. The automation bus has a Controller running automation programs, and a Web Video Server with cameras. Addressable components can be classified in two mayor groups: command modules and loads. The command modules are subdivided in two groups: those with human interface and those with sensors. Sensors are attached to the alarm bus, but their signals pass through the Interface into the automation bus. Command modules with human interfaces may have buttons, voice input and touch keys. We will refer to them generically as buttons. Controllers and automation programs are usually manufacturer specific to guarantee performance. Each automation program is a collection of scenarios. A scenario is a list of actions that happen based on precise list of events. For the controller, a scenario is a PLC (Programable Logic Control) instruction with the following syntax "When it happens: (event list) Only if: (expression = true) Execute: (command list)".

[*] The authors' home has a BTicino My Home automation system.

A program is a collection of scenarios created to solve a specific task, and its size is determined by the number of scenarios involved. The number of programs is an indication of the level of automation of a home that, like a computer, runs many programs simultaneously. Each program consists of many scenarios to make users' life comfortable, safe, and secure, as well as, providing energy savings.

Local Area Networks (LANs) are rapidly penetrating into homes, in the form of wired ethernet and Wi-Fi, to access information and entertainment services[2]. The Controller and the Web Video Server act as gateways between the Automation Bus and the LAN, so computers and iPod Touchs[3] have access to the Automation Bus. Additionally, the Web Video Server acts as a gateway between the Automation Bus and the internet to communicate with the Portal[4] for remote monitoring and control.

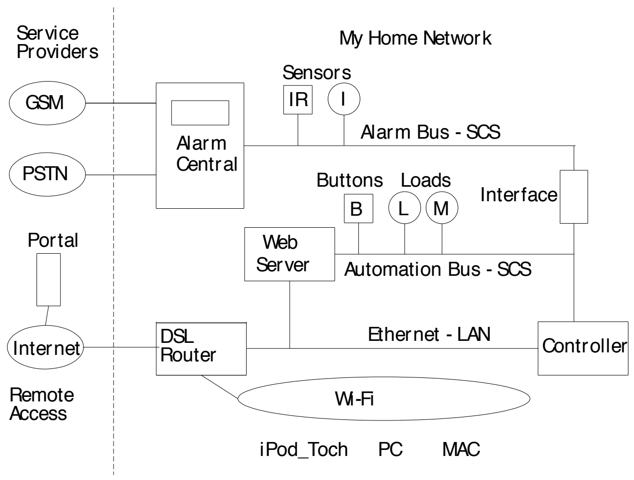

Fig. 1. My Home diagram for automation and security. The GSM network provides cellular access, the PSTN network provides telephone access, and the DSL router provides Internet access. The alarm central manages the devices on the Alarm Bus and the Automation Bus. The Controller manages only the devices on the Automation Bus. The Web Server manages the devices on the Automation Bus and the video cameras. It sends e-mails to the user about technical and intrusion events, and communicates with the Portal. The personal devices IPod Touch, PC and MAC communicate with the home controller to manage devices connected to the automation bus.

The words comfort and smart are widely used in HA, but they are not defined in measurable quantities. In the following paragraphs we will define Comfort, Smartness, and Performance as simple measurable quantities; also we will describe two novel programs to clarify these definitions and assess real world HA implementations.

2 How to Assess an Automated Home?

The size of an HA implementation can be precisely assessed by three counts: the buttons a user can press or touch, the addressable loads, and the scenarios stored in the Controller. This study proposes the mapping of the three counts into three ratios: Comfort, Smartness, and Performance to allows a better comparison and assessment of different implementations.

2.1 Comfort

2-way switches are popular in homes for controlling a load from several locations. Typical applications are in living rooms, bedrooms, stairs and hallways with two or more entrances. In these cases, one can define Comfort, C, as the ratio between the button count, B, and the load count, L, in the house. $C = B \div L$. In other words, C is defined as the average number of buttons per load.

2.2 Smartness

The Smartness, Q, is the ratio between the Scenario count, S, and the button count, B, in the house. $Q = S \div B$. In other words, Q is defined the average number of scenarios per button. It is important to note that Q does not depends in the number of loads in the house. Smartness is the ability of having many scenarios with few buttons. A traditional installation become smarter as time evolves by adding scenarios in the home controller to replace the functions of buttons.

A house with many buttons intimidates users because they are confronted with selecting the right button. Besides, it is expensive to add new buttons for each program. The Controller uses sensors, clocks schedules, and command sequences to increase Smartness by reducing the number of required buttons. The Home Controller monitors and uses the changes on the status of loads as events to trigger one or multiple scenarios,.

For example, think of the case of an elderly person that gets up from bed in the middle of the night to go to the bathroom. The Controller notices several motion events during the night and it turns on the lights in the proper sequence guiding the person from the bedroom to the bathroom, and in the proper sequence it turns off the lights off when the person goes back to bed. The person is never left alone in the dark, and, after the person goes back to bed, the lights are immediately turned off to minimize energy consumption. In addition, all without pressing a button, the Controller generates an auxiliary signal for the web server to send an e-mail saying the time and location of the event.

2.3 Performance

We define Performance, P, as the ratio: $P = S \div L$. In other words, P is the average number of scenarios per load. P can also be expressed as the product

$P = C \times Q$. It is important to note that P does not depends on the number of buttons in the house. Performance is the ability of a load to participate in multiple scenarios.

2.4 Our Home Example

Our home is wired with all lights and motors controlled by command modules with buttons. There are 100 buttons ($B = 100$), 69 loads ($L = 69$), and 67 scenarios ($S = 67$). This implementation results into $C = 1.49$, $Q = 0.67$, and $P = 0.97$. Let's note that $P > Q$. This means that the implementation gave more thought to reuse loads (due to the finite number of addresses available) than to reduce the number of buttons used.

Let's assume a one room house with only a light and a window shutter (two loads), three buttons (two for the shutter and one for the light), and nine programs for operation like: 1) presence simulation, 2) motion sensor, 3) distress signaling, 4) sun light control in the morning or afternoon, 5) morning alarm, 6) night scenario, 7) morning scenario, 8) room closed, and 9) room opened. Such a house has $C = 1.5$, $Q = 3$, and $P = 4.5$ at least, depending on the number of scenarios in each program. How can a room with a light and a window shutter may get a P that is larger than our house with many devices and buttons? The answer is hidden in the additional nine programs and sensors.

3 How to Program a Home?

It is not unusual to observe that modern houses built with smaller square footage require more thought to utilize the space than older houses with larger areas. The trend in construction has been to allocate an always smaller area per person, and increase the technology to make the space livable and functional. To increase Q and P requires to increase the number of scenarios. Thinking of new programs motivates the creation of scenarios that, once tried, may become part of our daily life. A challenge in the design of new homes is how to get users to customize their own scenarios. In future homes, controllers may learn from the user daily routines and adapt their commands to help the user.

Below, there is a list of the different programs implemented in our home using the Controller:

1. Open Home
2. Close Home
3. Good Morning (after waking up)
4. Goodnight (before going to bed)
5. Automated lighting (corridors)
6. External light control(porches, garages, and gardens)
7. Shutter position control (windows)
8. Simulated presence
9. Climate control

10. Hidden commands
11. Event sequence traps
12. Warning buzzer, panic call
13. Calendar schedule
14. Open door management.
15. Energy savings (no more that 5 lights at one time)
16. Elevator activation/deactivation
17. Room service
18. Person presence at the front door
19. Stairs light

We will describe only scenarios 10 and 11 from the previous list. "Hidden Commands" and "Event Sequence Traps" are programs written to increase Smartness and Performance.

3.1 Hidden Commands

A user may initiate a sequence of events by pressing one or more buttons to activate one or more loads. The Controller monitors the sequence of events and executes a sequence of commands. For example, when the nightstand light is ON, pressing twice (within three seconds) the button that controls a nightstand light causes the Controller to execute a goodnight sequence of commands that lowers all shutters and turns off all lights. Similarly, when the nightstand light is OFF, pressing twice (within three seconds) the button that controls the nightstand light causes the Controller to execute the good morning" sequence that raises all shutters in the house.

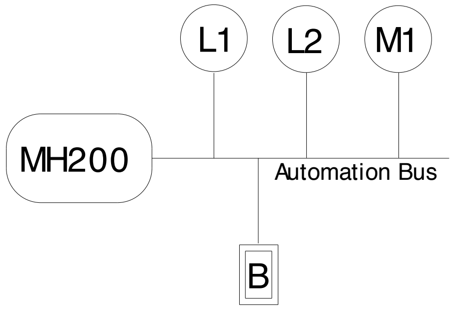

Fig. 2. Schematic of a BTcino My Home installation. $MH200$ is a controller, light $L1$, light $L2$, and Motor $M1$ are loads. B is a command module with a single button that normally toggles $L1$ ON/OFF. But, when button B is pressed twice within 3 sec., then one of two command sequences is executed depending on the initial status of $L1$: when $L1$ is ON, the shutter is lowered, when $L1$ is OFF, the shutter is raised.

Fig.2 shows the schematic of an HA installation using BTicino components to implement hidden commands. The installation consists of a button B, two lights $L1$ and $L2$, a motor $M1$, and a controller $MH200$, all interconnected by an Automation Bus. Normally, B is used to toggle $L1$ ON and OFF, but it may also be used to operate $M1$. Every time B is pressed, $L2$ is turned ON (by the MH200) for 3 seconds. If B is pressed a second time while $L2$ is ON, then the MH200 executes one of following two command sequences depending on the initial status of $L1$: when $L1$ is ON, the shutter is lowered, but when $L1$ is OFF, the shutter is raised. Below are the four scenarios for the MH200 to implement the program just described:

```
-L1 initially OFF:  #First time $B$ is pressed
   When: L1=ON Only_if: L2=OFF  #L1 went ON first time
   Execute: L2=ON, DELAY=3 sec, L2=OFF
-MORNING:  #Second time $B$ is pressed within 3 sec.
   When: L1=OFF Only_if: L2=ON   # L1 went ON and OFF within 3 sec
   Execute: M1=UP
-L1 initially ON: #First time $B$ is pressed
   When: L1=OFF Only_if: L2=OFF #L1 went OFF first time
   Execute: L2=ON, DELAY=3 sec, L2=OFF
-EVENING: #Second time $B$ is pressed within 3 sec.
   When: L1=ON Only_if: L2=ON    # L1 went OFF and ON within 3 sec
   Execute: M1=DOWN
```

Note that the MH200 responds to the changes of status of the load $L1$ and not to the signal from the button B. Finally, more commands can be added in the "Execute:" field to execute multiple actions.

According to our definitions of Comfort, Smartness and Programability, this HA implementation can be described as having $C = 1/3$, $Q = 4/1$, and $P = 4/3$. With $Q = 4$, this implementation is quite Smart! Q and P could further increased by programing additional scenarios that will raise the shutter automatically in the morning, close the shutter and turns L1=ON in the evening, and simulated presence, when not at home, by toggling L1 and L2 randomly.

3.2 Event Sequence Trap

This program implements a trap to set a panic call when an intruder fails to pass three tests after entering the house. The trap consists of switching three lights in the right sequence within a period of 20 seconds or otherwise, a panic call is placed. This program was implemented in the MH200 with the help of the finite state machine diagram shown in Fig.3. The transitions between states are equivalent to scenarios with conditions.

The program initiates in the START state, and after a trigger event it turns ON two lights, $L1$ and $L2$; and waits for the user to turn OFF $L2$. If the user turns OFF the correct light, $L2$, the program sets $L2$ ON again, moves to state TEST2, and waits for the user to turn OFF $L1$. If the user turns OFF the correct light, $L1$, then the program sets $L1$ ON again and moves to state TEST3 waiting for the user to turn ON $L3$. If the user successfully turns

$L3$ ON, the Controller blocks the panic call, returns to the START state, and waits for the trigger event. In this implementation the sequence of commands is L2=OFF, L1=OFF, and L3=ON. A different sequence within the 20 sec. delay period will not block the panic call . The idea of a trap was taken from the movie "Indiana Jones and Last Crusade" [5]. To set up one or more command sequences in a home or business may be quite useful to call for help in emergency situations. For example, a trigger is set every time a safe in opened and the user has to enter a right sequence of commands to disable a panic call.

The complete program is written below. Each scenario is equivalent to a state transition. In this program the trigger event is the $AUX9 = OFF$ that happens every time the alarm is deactivated.

```
-START-TEST1: When: AUX9=OFF #goto state TEST1
  Execute:  PANIC=OFF, L1=ON, L2=ON, C2=OFF, C1=ON, DELAY=20 sec., PANIC=ON
-TEST1-TEST2: When: L2=OFF  Only_if: C2=OFF AND C1=ON
  Execute: L2=ON, C2=ON, C1=OFF  #goto state TEST2
 -TEST1-PANIC: When: L1=OFF  Only_if: C2=OFF AND C1=ON
  Execute: PANIC=ON, L2=OFF, PANIC=OFF, C1=OFF  #goto state START
 -TEST2-TEST3: When: L1=OFF Only_if: C2=ON AND C1=OFF
  Execute: L2=ON, C1=ON #goto state TEST3
 -TEST2-PANIC: When: L2=OFF  Only_if: C2=ON AND C1=OFF
  Execute: PANIC=ON, L2=OFF, PANIC=OFF, C1=OFF   #goto state START
-TEST3-START: When: L3=ON Only_if: C2=ON AND C1=ON
  Execute: PANIC=BLOCK, L1=OFF, L2=OFF, DELAY=20 sec,
  PANIC=UNBLOCK, C2=OFF, C1=OFF  #goto state START
-TEST3-PANIC: When: L1=OFF, L2=OFF Only_if: C2=ON AND C1=ON
  Execute: PANIC=ON, L1=OFF, L2=OFF, PANIC=OFF,
  C1=OFF,C2=OFF  #goto state START
```

This implementation with 5 loads, 3 buttons, and 7 scenarios results in a $C = 0.6$, $Q = 2.3$, and $P = 1.4$. The reader interested in learning more about programming the Controller is encouraged to download and to read the MH200 Program Tutorial[6].

4 Summary

We have defined Comfort, Smartness, and Performance to compare diverse Home Automation Implementations. We have described home controller programs to provide the user with hidden commands and sequence traps to trigger scenarios for comfort and security applications. Hidden Commands could be very useful for users with restricted mobility with access to few buttons. By modeling scenarios as the transitions of finite state machines we were able to write sofisticated programs that involve several scenarios in one task. The addition of a home controller gives the HA implementation a high degree of flexibility by increasing the Programability and Smartness of a home.

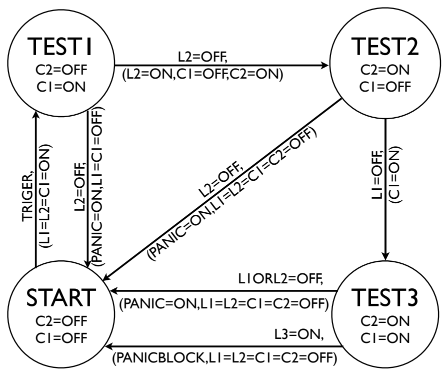

Fig. 3. Event Sequence Trap Finite State Machine Diagram. It consists of 4 states: $START$, $TEST1$, $TEST2$, and $TEST3$; 3 lights: $L1$, $L2$, and $L3$; and two loads $C1$ and $C2$ used to keep count of the states from zero to three. After a trigger event, the user has 20 sec. to pass the 3 tests and block the panic call.

References

1. BTicino catalog, `http://www.catalogo.bticino.it/catalog/images/it/bti/pdf/mh02g_1_gb_01.pdf`
2. Alice Home TV, `http://c5.telecomitalia.com/default.aspx?idPage=469`
3. `http://www.uptoweb.it/imyhome.html`
4. My Home Web Service, `http://www.myhome-bticino.it/web/myHomeWeb.page`
5. Indiana Jones and Last Crusade, `http://en.wikipedia.org/wiki/Indiana_Jones_and_the_Last_Crusade`
6. MH200 Program Tutorial, `http://www.myhomevillage.com`