Task partitioning in smart environments

Markus Ritala
Tampere University of Technology
Institute of Electronics
P.O.Box 692, 33101 Tampere, Finland
+358 50 482 1336
ritala@ele.tut.fi

Jukka Vanhala
Tampere University of Technology
Institute of Electronics
P.O.Box 692, 33101 Tampere, Finland
+358 3 3115 3383
jukka.vanhala@tut.fi

ABSTRACT

This paper studies the use of a Bluetooth enabled mobile phone as a controlling device in a smart environment. Main focus for the work is partitioning of the functionality and the UI handling between a mobile phone and a smart appliance. There seems to be no single right solution for the partitioning of functions. Instead, there are so many case dependent characteristics that the partitioning of functions should be designed in a case-specific fashion. The results of this work give the basis for designing the partitioning.

Keywords

Smart Environment, Distributed UI, Software Architecture, UI Management System

1. INTRODUCTION

The number of smart environments is increasing rapidly [1] and despite the embedded intelligence the users have to be able to control their environment also manually. Although the level of embedded intelligence is increasing, the principal goal should be to activate the user and not to let the home automation take over controlling everything [3][8]. The capabilities of mobile phones have improved significantly in the past few years. Therefore the so-called smart phones offer an interesting opportunity for controlling a smart environment.

It is clear that when there is some smart environment system, there will be appliances from many different manufacturers and those appliances should be able to communicate with each other. They must have the same communication medium, protocols, and semantics of the communication for all the appliances. In addition, when designing the architecture of the control system for the smart environment, an important question is which part of the system has the responsibility for handling which tasks. Here we concentrate on the task partitioning of the User Interface (UI) handling and the functionality of the appliance between the smart environment and the UI device. A Bluetooth (BT) enabled mobile phone is used as a UI device. Also the limitations set by the communication technology are considered. This study has been carried out as a part of the smart environment research at Tampere University of Technology, Institute of Electronics [3]. The results are based on both theoretical considerations and the practical implementations.

At the moment there are two smart home implementations used in this study. The newer one, called eHome [5][6], is a normal two-room flat in an apartment house with paying tenants. The other is a laboratory installation located at the campus and called the Living Room [6][8]. The Living Room includes a hall, a bedroom, a living room, a dining room, a kitchen, a WC, and a sauna. The environment is equipped with several smart appliances from several manufacturers. They are all connected to the controlling system via a cable or a wireless radio link. There are also several different kinds of parallel UI systems [9][10]. New appliances and UI systems can easily be installed despite of the dynamic and distributed nature of the environment.

2. TERMINOLOGY

Before we can discuss about different partitioning solutions, we need to clarify the objects we are talking about. Every *smart appliance* has its characteristic function and it performs *actions*. The abstraction level of these actions may vary a lot. The actions from upper abstraction level may take advantage of actions from lower abstraction level to complete its own tasks.

As an example we may consider a door. Its function is to block the doorway and open to let someone trough. There may be upper abstraction level actions like “open the door” and “close the door”. There may also be actions from lower abstraction level like “turn the door outwards”, “turn the door inwards”, “is the door open”, “is the door closed” and “stop the door”. These lower abstraction level actions could be used to perform previously mentioned upper abstraction level actions. As a result of this, the action that the user thinks as a one simple action may actually be a sequence of many smaller actions. The user usually cares only about actions on the uppermost abstraction level whereas the lower level actions map to hardware capabilities.

In this paper, a collection of those actions that a single smart appliance is capable of performing are referred to as its *functionality*.

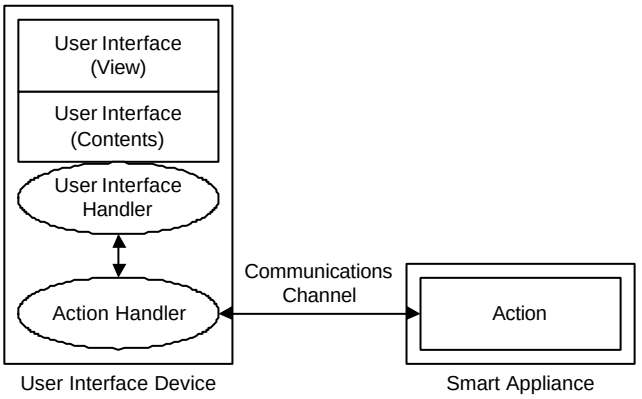

Figure 1: Objects of the control system from upper left corner of the table 2.

Every smart appliance has to offer some way to use its functionality. This is the UI of the device. Here it is referred to as *control interface* to avoid confusion. In an example we used before, the control interface could have two commands. By sending one of

those to the door, it will open and using the other the door will close. As we can see, the UI of the appliance does not need to be visible for the human user. It may be for example a list of commands and parameters that are sent to the device trough a communication channel (figure 1).

There may also be description of the view that is meant to be shown to the user. Because the description of the view is not exact, the actual view that the user sees in a controlling device may vary a lot depending on the type of the UI device.

In this paper, the UI is the description of the methods to use the functionality of the certain smart appliance. A *view* is the representation for the user of the UI on some UI device. In the Figure 2 is an example of UI of few appliances. There is also the representation of the UIs, the views on Nokia 7650 mobile phone.

Every UI needs also the logic for handling the UI. The UI handler has to maintain the current state of the UI. Because the screen on a mobile phone is rather limited, in the UI there should be only those functions and parameters that are active for use at the current moment. In this paper the *UI handler* is the part of the system, which can keep the UI updated and display the current UI whenever it is needed.

```
<LIST_ITEM>
2
3
Lower curtain
00 10 60 29 06 BD 01 A0
<END_LIST_ITEM>
<LIST_ITEM>
4
5
Lights on
00 50 37 F9 9D 12 03 A2 0F 04
<END_LIST_ITEM>
<LIST_ITEM>
6
7
Open television
00 02 78 03 F2 C4  01 A2 0F 03
<END_LIST_ITEM>
<LIST_ITEM>
8
9
Close door
00 0E 57 99 A6 D5  06 A0
<END_LIST_ITEM>
```

Figure 2: The view (right) generated from the UI (left) on the Nokia 7650 mobile phone.

3. PARTITIONING OF TASKS

While partitioning the functionality and the UI between the UI device and the smart appliance, there are four situations. It is possible to set both of these tasks for the responsibility of the appliance or the mobile phone. Again, it is possible to divide the responsibility between these parts of the systems such as the other one of those has a responsibility for handling the UI issues and the other one handles the functionality issues.

3.1 UI handling in the smart appliance

When the UI and the UI handling are located in the smart appliance, every time when there is need for the UI information it has to be asked from the device itself. This increases the amount of data that is needed to be delivered in the controlling system. Consequently the need for communication speed is also increased. In addition, there have to be enough resources on the appliance for handling and storing the needed data. Objects of the control system and the interaction between them are shown in figure 3.

For single controlling task there might be several situation where the UI and the description of the view has to be delivered into the controlling device. For example, when we are browsing a menu, the entire user input has to be delivered first into controlling device. The controlling device performs the needed tasks and returns the new UI description. This has to be done in that way because in this solution the controlling device does not know anything about the context of the command. It only acts as a physical UI device and delivers all the user actions ahead. This approach could be called the dumb terminal approach.

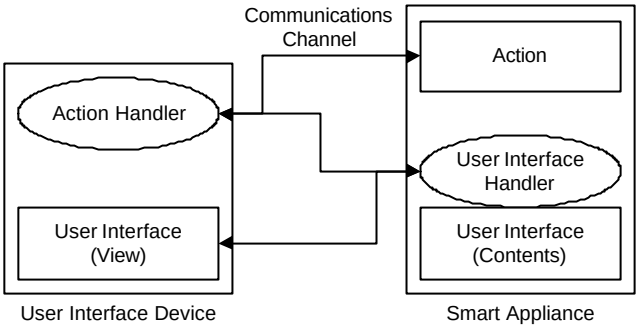

Figure 3: Objects of the control system from upper right corner of the table 2.

A good example of this kind of UI handling system is the prototype of the context aware controlling system made at [11]. In this prototype, the controlling device offers only the physical UI for the user. All the system information is located in the centralized controlling server. The server has responsibility of the UI handling and delivers the current description of it to the controlling device.

3.2 UI handling in a mobile phone

When the responsibility of the UI handling is in the UI device e.g. a mobile phone or a PDA, all the actions that have an influence only on the UI may be done locally in the mobile phone. This solution decreases the amount of transferred data compared with the previous situation. That might be very significant property when the smart environment has several smart appliances and the amount of transferred data is increasing.

Because the computational capacity of the mobile phones is developing rapidly, it is reasonable to centralize the UI processing into these devices. As a result, the complexity and cost of the appliances is lower.

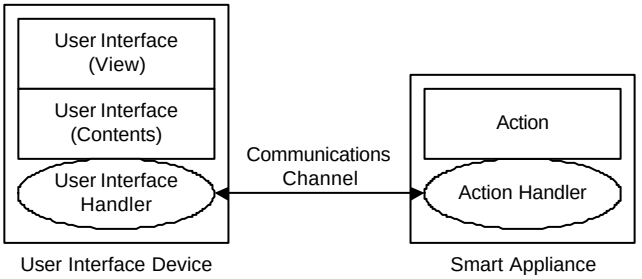

Figure 4: Objects of the system from lower left corner of the table 2.

Centralizing functionality means that the controlling system is much easier to maintain. All the updates are easier to make only to a mobile phone, because the amount of controlling devices are usually lower that the amount of appliances. Also the software development is easier, cheaper and faster for mobile phones than for embedded systems.

It is also possible to create the views into mobile phone that the single smart appliance cannot offer, for example when the function uses several separate appliances to complete its task. This situation from the users point of view is very difficult if appliances handle the UI. Also the location information of the appliance can be easily configured into the mobile phone.

3.3 The functionality in a smart appliance

Setting the functionality into the smart appliance means, that it performs complicated tasks independently using its own basic functions. This partitioning decreases the need for communication between the appliance and the mobile phone. For each controlled task only one or very few control commands need to be sent to the appliance. A single controlling command is very short compared to even a simple UI description.

Again, when increasing the functions of the smart appliance also the complexity of the device increases. This increases the size and of course the cost of the device (figure 5). Secondly it is very hard to combine the functions from separate appliances to perform some more advanced task because the controlling device is not capable to make any decisions and interaction by itself.

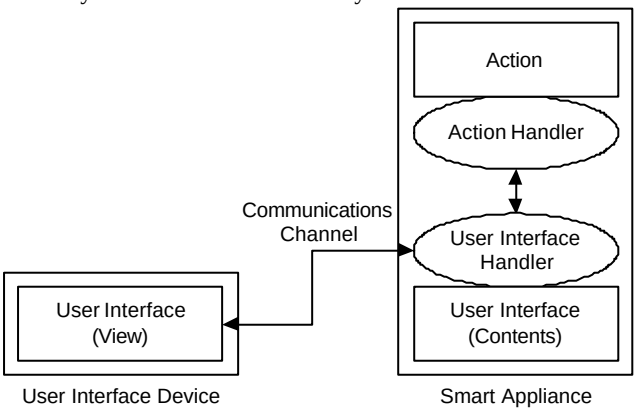

Figure 5: Objects of the control system from lower right corner of the table 2.

However, it is easy to import a new smart appliance into current smart environment because the device includes all the needed functions to work and there are no other requirements for the environment.

3.4 The functionality in a mobile phone

If the mobile phone acting as a controlling device has a responsibility of the functionality of the controlled system it means, that it will perform all the tasks by using the basic functions of the smart appliance (figure 1). As an example we might have a situation where the user wants the car to be heated when he/she is planned to go to work at 7:30 am. In this situation the mobile phone could use an outdoor temperature sensor and a real time clock with the car heater to perform this task. In the Figure 6 there is a simplified graph of the situation. This kind of system uses more communication services than the previously presented system. But because these controlling commands are quite small, the total amount of transferred data is not as large. In spite of that, one controlling situation may need many controlling commands to be delivered to the separate devices and those commands come usually in short bursts, so the communication system should be able to deliver all these command with no noticeable delay.

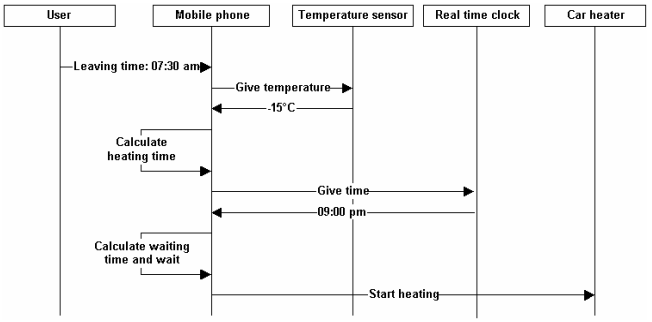

Figure 6: The mobile phone is performing a complicated task by using basic functions of several separate smart appliances.

When we are using the BT as a communication channel, the capacity of the communication will not be the bottleneck with controlling commands. The problems come when there is a need to make a connection between the mobile phone and a smart appliance and after a few commands disconnect and make a connection to another appliance. There can be noticeable delays in connecting and disconnecting the BT devices. In this situation it is important to decide when it is the right time to disconnect.

Partitioning the functionality to the mobile phone, it is possible to take advance of all the mobile phones resources. When the situation is complicated with several devices to control to get the tasks performed, the need of computing and storage capacity increases rapidly.

4. PRACTICAL SOLUTIONS

In this study a simple demonstration of the three out of four possible partitioning situations are implemented (excluding the case in figure 3). Using this demonstration it was easy to find out the characteristics and differences of the situations. The smart appliance used is a simple controllable blinder in the smart environment described earlier. The controlling software was made for the Nokia Series 60 SDK.

Table 1: The controlling commands for controllable blinder.

Function	Description
Blind down	Moves the blind into lower position
Blind up	Moves the blind into upper position
Blind downwards	Moves the blind downwards until it is stopped using separate command
Blind upwards	Moves the blind upwards until it is stopped using a separate command
Stop blind	Stops the blind
Is blind in low position	Tells if the blind is in the lower position
Is blind in up position	Tells if the blind is in the upper position

The blinder can be controlled via a BT connection. A mobile phone acts as a controlling device. The blinder can be set only to upper or lower position. The controlling commands are presented in the table 1. The first two actions are on the upper abstraction level.

These actions are the ones, which the user wants to be completed. These two tasks are actually performed by using the lower level commands.

In the figure 7 there is the construction of the controllable blinder. In the sides of the window, there are guide bars to guide the blind. A magnetic switch tells if the blind is in the lower position. The blind controller and the BT module are behind the window, out of sight.

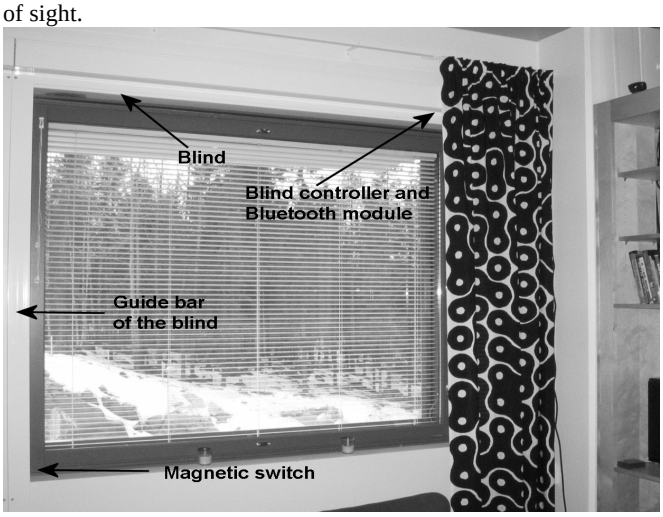

Figure 7: The components of the controllable blinder.

5. CONCLUSIONS

From the further development point of view, placing the responsibility of functionality to the mobile phone brings up many interesting possibilities. One small function of some new appliance may make it possible to implement complex actions. The small sensors like humidity and temperature sensor are the first appliances in a smart environment. When the number of small appliances increases, the new more complex controlling tasks become available by combining these smaller one. Soon only the imagination sets limits for that.

Table 2: characteristics of different partitionings

Functionality \ User Interface	Mobile Phone	Smart Appliance
Mobile Phone	+ Phone characteristics usage + Versatile user interfaces + Combining functions - Sensitive for phone damages - Difficult to add smart appliances	+ Easy to expand + Robustness - Amount of communication - Possible delays - Slowness of connection creation - Hard to implement
Smart Appliance	+ Amount of communication + Easy to control + Easy to implement - Difficult to expand - Difficult to update	+ Clear entirety + Robustness + Easy to add new functions - Complexity of smart appliances - Prize of smart appliances - Limitations of functions

In the Table 2 the characteristics of different partitioning situations are summarized. In the horizontal axis the partitioning situation from the functionality point of view is described and in the vertical axis there are both the UI partitioning situations. This table illustrates how we can in most cases find very different characteristics from the opposite sides of the table.

Partitioning situation in the upper right corner of the table is very difficult to implement in a practice. If UIs are handled in the smart appliance and all the complicated controlling tasks are performed by the mobile phone, it is very hard to create the view for the user. Because a single appliance doesn't know anything about the total controlling situation that its basic function is used for. Therefore in practice, if the functionality is on the mobile phones responsibility, also the UI handling has to be there too.

There is no single right solution for the partitioning of functions. Instead, there are so many case dependent characteristics in every control system and environment that the partitioning of the functions should be designed in a case-specific fashion. The results of this work give a very good basis for designing the partitioning. The study also brings up some issues that could be considered more thoroughly in a further investigation.

The extreme partitioning situations described here above are probably not the best possible for smart environment controlling system. It might be reasonable to use some partitioning between them. That's how we usually get the most suitable characteristics for the system in many cases. For an example we may consider a situation where a UI can be downloaded from the smart appliance using BT. After that the mobile phone has a responsibility of controlling the view and combine the functions of the appliance to get more advanced tasks performed.

6. REFERENCES

[1] Lehto, M., Tekniikkaa ikä kaikki, Finnish Environment 244, Ministry of Environment, Housing and Building, Oy Edita Ab, Helsinki, 6/1998, ISBN 1238-7312

[2] Kaila, L., The Intelligent Home of the Future, Diploma thesis, TUT Electronics, 09.05.2001

[3] Mikkonen, J., Kaila, L., Vanhala, J., "Living Room –project: Towards User Activation", Proceedings of the Interfacing Reality in the New Millennium, OZCHI 2000, Sydney, AU, 12/2000

[4] TUT, Personal Electronics Group, [online, 19.05.3002], URL: http://www.ele.tut.fi/research/ personalelectronics/

[5] TUT Electronics, eKoti, [online, 31.01.2004], URL: http://www.ele.tut.fi/research/personalelectronics/projects/ ekoti_03/index.htm

[6] Koskela, T., Smart Home Usability and Living Experience, Diploma thesis, TUT Software Systems, 04.06.2003

[7] TUT Electronics, Living Room, [online, 26.01.2004], URL: http://www.ele.tut.fi/research/personalelectronics/projects/ smart_home.htm

[8] Kaila, L., Mikkonen, J., Keski-Opas, P., Vanhala, J., "Living Room", Proceedings of the Second International Symposium, HUC 2000, Bristol, UK, 8/2000

[9] Hyvönen, J., Speech control system for intelligent environment, Diploma thesis, TUT Electronics, 28.02.2003

[10] Lehti, L., Älykodin matkapuhelinohjaus, Diploma thesis, Tampere University of Technology, Institute of Electronics, Tampere, 15.01.2003

[11] Ritala, M., Tieranta, T., Vanhala, J., "Context Aware UI System for Smart Home Control", Proceedings of the HOIT 2003, Irvine, USA, 06. – 08.04.2003