Designing Embedded System for Handheld Mobile Cell Phones for Efficient Remote Control of Household Electronic Devices

By

Onukwugha C. G.

Department of Computer Science, University of Port Harcourt

and

Asagba P. O.

Department of Computer Science, University of Port Harcourt

Abstract

Smart spaces have interactive and intelligent capabilities that the home can use to adaptively control many aspects of the environment such as climate, water, lighting, maintenance and multimedia entertainment. Among the numerous benefits inherent in a smart home is that a smart home can save money through savings in heating, lighting, air conditioners, etc. This paper presents the software and hardware design of embedded system for handheld mobile Phone remote control of electronic household appliances.

Keywords: Embedded system, smart spaces, mobile phones, pervasive computing, energy conservation

1.0 Introduction

The current trend in telecommunications is that virtually everybody, young and old, rich and poor, has mobile phone. Mobile handheld devices such as smart phones and personal digital assistants (PDA’s) are extremely popular these days. People carry handheld devices anytime, anywhere and use them to perform daily activities like making calls, checking e-mails and schedules, and browsing the mobile web.
According to several market research reports, the sales of mobile phones, smartphones and PC’s (including desktops, servers, and laptops) were 291.6, 32.2, and 69.5 million units in Q1 2008, respectively (Hu, 2010). The current trend in computing comprises a lot of portable devices which are also cost effective. The sensors, mobile devices, system-on-a-chip, etc are all effortlessly portable and make available cheap computing power. At their core, all models of ubiquitous computing share a vision of small, inexpensive, robust networked processing devices, distributed at all scales throughout everyday life. The proliferation of these numerous computing devices and computers has given rise to the building of environments that are capable of being controlled remotely because the devices that are in such environment are networked.

An environment that is made up of electronic and computing devices that are connected in such a way that they interact over a wireless network is called a smart space or smart home (Wiki, 2012). “Smart Spaces” are ordinary environments equipped with visual and audio sensing systems that can perceive and react to people without requiring them to wear any special equipment. Mark Weiser’s (Weiser, 1991) dream of calm, disappearing and ubiquitous computing cannot be discussed without the mention of smart spaces. A smart environment or space is a region of the real world that is extensively equipped with sensors, actuators and computing components (Nixon et al, 1999). In effect the smart space becomes a part of a larger information system with all actions within the space potentially effecting the underlying computer applications, which may themselves affect the space through the actuators. Such smart environments have tremendous potential within many application areas to improve the utility of a space. Smart home enables centralized control over lighting, HVAC (heating, ventilation and air conditioning), appliances, and other systems, to provide improved convenience, comfort, energy efficiency and security.

Since each of the numerous electronic appliances we have in the home has a remote control, it is obvious that clustering a room with many “remote controls” could be inconveniencing and frustrating sometimes (Johnson, 2004). This is because for one to switch off the air conditioner, for instance, in the living room, one has to go to the living room to begin to search among all the ”remote controls” in that room. This
exercise amounts to so much inconvenience. At some other times, we could forgetfully or willfully (as a result of tiredness) allow the electronic devices to be on when actually nobody is using them, energy which would have been conserved is wasted.

In this paper we present how we can work out a way in which the mobile cell phone we carry will serve as a universal LED remote control for the networked electronic devices in our homes. A lot of energy will be conserved, convenience and comfort will equally be achieved since the user do not need to be at the point of the device or in the home where the appliances reside to be able to control the device .

2.0 Context-Driven Task-Oriented Middleware

A great number of research works have been carried out in the area of smart space development. The CDTOM (Context-Driven Task-Oriented Middleware) for pervasive homecare environment (Hongbo et al, 2011) is a smart home which can assist inhabitants (users) to live more conveniently and harmoniously. The work proposes a context-driven task-oriented middleware to meet the challenge of perceiving the environment to access occurring situations, thus allowing systems to behave intelligently according to user intent. The main strength of CDOM is its separation of tasks and services which would allow for greater flexibility in changing the tasks without changing the services and vice versa, and also hiding the complexity of composing various services in pervasive environment from the users. CDTOM is able to deliver applications that could assist end-users perform their intended tasks in an automatic and adaptive manner. Event-Condition-Action (ECA) mechanism is adopted in implementing the mapping between actions and services.

The Ubiquitous Home (Hu, 2010) presents a system for retrieval and summarization of continuously captured multimedia data from a two-room house consisting of a large number of cameras and microphones. The Ubiquitous home (Yamazaki, 2005) has been designed to provide a testing ground for ubiquitous sensing in a household environment. Simulating a two-bedroom house, it is equipped with 17 cameras and 25 microphones for continuous acquisition of video and audio. Pressure-based floor sensors are mounted in a rectangular grid.
The MavHome (Managing an Intelligent Versatile Home) smart home project is a multi-disciplinary research project at the University of Texas at Arlington focused on the creation of intelligent and versatile home environment (Cook et al, 2003). MavHome combines technologies from Artificial Intelligence, Machine Learning, databases, Mobile Computing, Robotics, and Multimedia Computing to develop a smart home that acts as a rational agent that has the ability to record inhabitant interaction and trigger sequences of events. MavHome tested several prediction algorithms: Smart Home Inhabitant Prediction (SHIP), Active Lezi, Task-based Markov Model (TMM), Episode Discovery (a data mining algorithm). Drawing from the strengths of the various prediction algorithms, MavHome came up with the unique prediction algorithm called Predict.

3.0 Application Development by the User

Application development refers to the process of building applications by assembling services offered by computing devices embedded in the environment. Figure 1 shows the generic application process.

![Fig.1: Generic Application Composition Process. Courtesy: Davidyuk et al (2009)](image)

As shown in figure 1, the application composition process involves two separate actors, the end-user and the service provider. The role of the service provider is to develop and publish services which provide some functionality through a well-defined
interface (Ben, 2007). The end-user’s role is to create applications and utilize (i.e. interact with) applications (Davidyuk et al, 2009).

The application composition process relies on descriptions of applications and services. An application description specifies the abstract services the application is composed of and the relationships between them (i.e. control and data flows). The application composer is responsible for decision-making in the application composition process. The application composer uses application descriptions and possible composition criteria (preferences, fidelity constraints, cost, etc.) provided by the end-user as inputs and chooses available services which satisfy the given criteria. The selection of services is supported by a service discovery protocol, which is responsible for the matchmaking functionality (i.e. for matching service discovery request against the service descriptions stored in the service registry). Since the discovered set of services may potentially contain redundant instances, the application composer optimizes (i.e. further reduces) the service set and produces an application configuration (i.e. application composition plan), which satisfies the criteria given earlier. After this, the application is instantiated and executed by the run-time environment. During execution, the application can be adapted (i.e. recomposed) according to user defined adaptation policies. These policies are formed rules which trigger actions when the application or user context changes.

Several design styles for developing adaptive ubiquitous applications through composition have been suggested (Paluska et al, 2008).

4.0 Design method for the Smart System

System design involves the evaluation of alternative solutions and the specification of detailed computer-based solution. The key designs here are object-oriented design and procedural or structured design. Whereas the analysis focuses on the logical, implementation independent aspects of a system, design deals with the physical or implementation dependent aspects of the system.

In design, the object-oriented approach involves the classification of the entire system into classes that are capable of interacting with one another. These classes are
then abstracted and designed in such a way as to build the system in a conceptual form. The structured approach on the other hand is a process-oriented technique for breaking up a larger program into a hierarchy of modules that results in a computer program that is easily implemented and maintained.

4.1 Object-Oriented System Design

Object-Oriented Design involves the domain design of the system use case using manual or Object-Oriented Computer Aided Software Engineering (OOCASE) tools. The OO design process is evolutionary and iterative. During OOD the designer may need to revise the data or process characteristics for an object that was defined during the system analysis. Design implementation decision may necessitate that the designer define a new set of objects that will make up an interface screen that the user may interact with in the new system. It encompasses abstraction of the system attributes and behaviour and using the necessary tools in building their design. Unified Modeling Language (UML) is often used for object-oriented design process. Unlike the structured strategy, object-oriented paradigm considers both data and actions to be of equal importance. An object is a unified software component that incorporates both the data and the actions that operate on the data. Hence, data is encapsulated in the action of every class, object and message. Objects are identified in real life, classified and defined in terms of attributes and operations. Object are identified in software from the problem statement by underling each noun or noun phrase and entering it in a simple table. Then the process of elimination is used to whittle this list of all nouns/noun phrases into a set of appropriate classes. These classes are then used to automate the various actions within the system giving way for the system to be implemented. The appropriate tool or programming language need to be used in handling the actions within the system. This is necessary due to the fact that object oriented designs need object-oriented coding to be able to follow the specified action stated on the design document. In some cases more than one programming language may be needed. One for the coding of the processor and the other for the coding of the mobile device.
4.2 PC-Space Model

The UML enables us to model, via class diagrams, the classes in the PC-space system and their relationships. In UML, each class is modeled as a rectangle with three compartments. The top compartment contains the name of the class, the middle compartment contains the class attributes while the bottom compartment contains the class operations.

4.3 Process Model and Design

In this project we will use the methods defined in the http request response model for mobile and smart systems to design the process that the system will deploy in executing the process of the system to be developed in the next chapter. In figure 2.1a the design shows the present Smart space. Users send digital signal to the smart space environment as a one-to-one model.

In the present system therefore the user needs to send information to the gadgets in the smart space one item at a time to respond in the normal way a single mobile device communicates with a single mobile device.

In our own proposed model in figure 2.1b, the system is designed and improved upon so that a smart space user can send digital signal to the smart space environment as a one-to-many model. This will allow many gadgets to be controlled in the smart space using single message or signal. This is an improvement since single signal can be passed to the entire gadgets on the smart space using only a single signal from any remote location.
This multi-reception is made possible by the introduction of Embedded Engine responsible for processing in the Device message sent by the user device. The user may do the sending of signal in the same way but the embedded engine interprets the message targeting different gadgets within the smart space. The proposed model also have a litecode module to handle archived and used messages that can be useful in recording user activity in the smart space. This will enable the smart space administrator to have information concerning the activities within the space and who in the house is actually doing what within the space. Special message is also configured to be interpreted for single gadget so that a user that has exclusive control
of the gadget can manipulate the gadget without interfering with the others as illustrated in figure 2.1b.

![Diagram of the proposed smart space process design]

**Fig 2.1b : Proposed Smart Space Process Design**

The digital request is batched and sent into the smart space engine for processing before it can be forwarded to the device where the embedded system processes the request made by the user of the smart space. If the request requires the use of any message archive stored in the system for the purpose of processing the request it
interacts with the mobile device through stored Litecode data which it consequently uses in the process of handling the system. Once the embedded system is triggered the system automatically responds to the users request. The users could also make request to examine the operation of the system to verify if the response is actually the one required from the system. The smart space system will in turn return response based on the activity that the user has sent via the mobile device within the smart space environment.

4.4 CASE Design of Electronic Interface

The electronic interface is the hardware component design that illustrate the schematic features and interface of the entire system. A computer aided system engineering tool-Simulink was used in designing the electronic interface. It clearly shows the manual switch equivalent of the message responses expected for each activation of the lead. The message code is also shown in the design and its corresponding actions expected from the system. In the design a black lead illustrate a system lead that is OFF while the Green lead illustrate a system lead that is ON. Figure 3 clearly illustrates all of that showing were the lead is switched ON or OFF.

![Fig 3: A CASE Electronic Interface Design](image-url)
4.5 The Circuit Design

The design of the circuit is based on the usual voltage divider equations to obtain the required current and voltage specifications for the Smart home circuit. Figure 4 shows the smart home transmitter schematic design.

![Smart Home Transmitter Schematic Design](image)

Fig 4: Smart Home Transmitter Schematic Design

In the diagram of figure 4 the transmitter schematic design is presented. This system receives signal from the user via the user interface and transmits the interpreted diagram into electronic format that can be received and interpreted by the receiving circuit on the receiving gadget within the smart space.

In figure 5, the electronic design of the receiver is equally presented. The receiver gets the electronic signal sent from the transmitter and translates it into the corresponding triggers which the electronic gadgets within the smart space environment will execute or react to.
Fig 5: Smart Space Receiver Schematic Design

The receiver uses the wireless technology to accept the signal presented to it via the signal points in both the transmitter and receiver. The signal point is implemented as sensor points that are capable of accepting communication signal within a radio signal radius which is always dependent on the signal strength and the area of coverage of the smart space.

5.0 Implementation

Java is the chosen language for the implementation of the Smart Space System (SSS) because of its platform-independent nature and rich Application Programming Interface (APIs) for mobile devices like Nokia (Nokia, 2009). Within the java technology, there are different programming language packs each suited for specific tasks. On the standalone machine Java Standard edition (JSE) is more suited for the development. On wireless and other mobile devices Java Micro Edition (JME) is more
suited with its MIDP and MIDlet application. JSE can also be used if the deployment is on the android mobile devices. In achieving the complete implementation, we have combined JSE and JME as the two core programming languages within the Java platform used for the development of our application for the implementation of our model. The code generated from the programming languages come in Java Archive (JAR) format which is usually an executable byte codes that executes within the Java Run time environment both at the personal computer level for JSE and in Java Archive Deployer (JAD) format which is usually executed byte code in Run time environments on mobile devices for JME.

The proposed model was implemented using NetBeans 7.0 integrated development environment. The whole program was implemented in about five modules viz, the Login (authentication) module, transmission module, receiving module, verification module and triggering module. The authentication, transmission and verification processes take place at the sending point while the receiving and triggering process take place at the receiving point. The smart space system needed to be tested and deployed after writing and debugging the program, testing is done to assess the efficiency of the program.

5.1 System Deployment

After testing the system, we deployed them in the various areas where the application is supposed to be executed. The stand alone system needs to execute on PC while the MIDP is executed on mobile devices such as phones and PDAs. The deployment in all these environments is to make sure that all the environments where the smart space system will execute are verified.

6.0 Conclusion

The use of mobile phone as an efficient universal remote control is based upon the fact that virtually everybody has one or more mobile phones. In this paper, we have shown that with embedded system technology software application built into a cell phone (Java enabled) can be used to monitor and control home electronic devices. This is an additional feature to the other things that mobile phones are used for. One signal from the phone can control one or many devices according to user intent. In
effect, this paper has introduced an easy and convenient way of controlling networked electronic devices.

References


Wikipedia_6Wikipedia the free encyclopedia (2012)