Name: Srikanthan Parameshwaran

MCS Sentinel Home Project Plan

(March, 2007)

Academic Mentor:

1) Dr S Manoharan

Industry Mentor:

1) Sean Tindale

Introduction / problem:

Our aim is to build a system that use smart technology and digital convergence to integrate and automate every aspect of our home or office, from telephones to computers, security systems, intercoms, entertainment systems, lighting, curtains, blinds, spas, swimming pools and whatever else you like. As long as it switches on and off it can probably be done. The system will have ability to combine individual services into one package creating an internal network where all your appliances, amenities, facilities and services talk to each other. For now the aim is not to integrate individual systems but to keep each system separate and control them individually. [2]

The aim is to control the system from anywhere in the world using text message and the internet. The user can send a text message to the server to get information about the current stat of the environment its monitoring, or to give a command to the server, e.g. turn on the alarm. The system will also use txt message or email to inform the user if there is any problem in their house or office.

Why have the system?

Let say you and your family is half way to the beach and you wife goes I left the oven on or you realize that you didn't turn on the alarm system or your child goes I left the heater on. The only option you have is to go back home fix this problem as you neighbors are not at home. This is waste of time and money to go back and fix the problem. One option would be to enable you to use your mobile to control all the things in the house so you can turn on/off applications whenever and wherever you were.

The other problem could be that you are in the wine business and to product top quality wines, you need top quality grapes and to product top quality grapes, the grapes plants should have the optimum condition. When I say optimum condition, the soil should have the correct PH or the temperature should be in the correct range and ….. To provide optimum condition to the grapes plants you can have someone in the vineyard who keeps eye on the condition or have a system that does this and inform you when there is a problem. Having a system is cheaper as most of the time the conditions will be optimum.

The system can be set to do certain task automatically based on periodically or on the current situation. One example of periodically could be that, turn on the gardening water system at 6PM everyday and one example of current situation could be that keep the temperature of the house or the office at a certain level.

Having this system not only gives you piece of mind that your house or office will last well into the future - safe, secure, healthy, energy efficient, cost effective, high quality and manageable, but it also provides the convenience of being able to control any aspect of your home whether it be from the room next door or from your office located across town. [2]

My requirement is to make the mobile interface that will enable the client and the server to communicate, and to make the communication secure and efficient as possible.

The problem with using txt message (ASCII) to communicate with the server is that each txt message must be no more than 160 characters. When you send multiple commands in one txt message, you will almost certainly reach the cap of 160 characters. My job is to do some research and find smarter ways of sending

message that enable the user to send multiple commands in one txt message. The one option is to use binary rather than ASCII and the other option could be to use compression on the ASCII message send or use compressed binary. My job is to find the best option for our project and implement that.

Solutions:

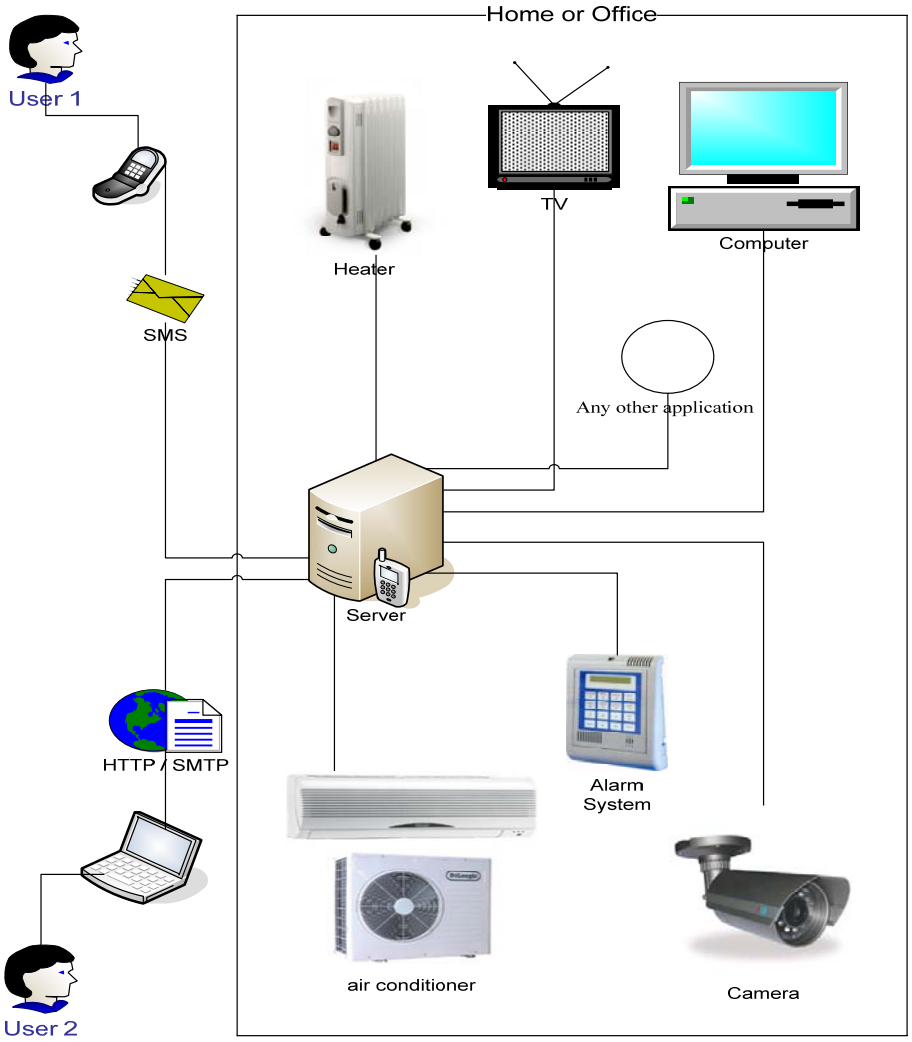


Figure 1: System One

Figure 1: Shows one option of how to implement the system and how the user communicates with the applications. The above picture shows that user can use SMS or HTTP to talk to the server and server can use SMS and SMTP (email) to communicate with user.

The architecture above requires only one public IP address and only one mobile number (SIM). Users or any application that wants to talk to an application has to communicate through the server; this server is like the center brain of the system. This brain controls the home or office integration. The communication between the server and the applications can be through the wires or wireless.

If we planed to use the wire as the medium, we have 3 options:

1) Put new wiring like LAN cables to connect all the applications.
2) Use the existing power wires to communicate
3) To use the existing phone lines

Each option has its advantage and disadvantage and also the choice is depended on the availability and the affordability of the correct hardware components to do this. For example, if you are building a new house or office, it will be good to put a separate wiring for the system as it will be secure and cheaper, but if the system is to be implemented in an existing house or office; it would be cheaper and easy to use one of the other option or to use wireless.

Using wireless (IEEE 802.11g or similar technology) to communicate between the server and the applications is a good option except the fact it's less secure and more expensive. Security can be almost fixed using encryption but wireless are normally expensive than non-wireless.

There are companies like, HAI (Home Automation, Inc.) or X10 and many more providing similar products. The problem with their product is that they have

separate systems for everything, e.g. one system for home security, another one for web monitoring. The other problem is that the quality is cheap and the price is high. Our system will provide high quality product with affordable price.

Our aim is to integrate every single application in the house or office and allow them to talk to each other and to reduce the amount of configuration the user has to make. The aim is to build a system like plug and use, this is when the application is turned on and is plugged into the network, the server and the application will communicate automatically and exchange information about each other. Some examples of the information could be that the application will let the server know about what type of application is it, e.g. TV, Computer, …, and server will issue an id to the application and this id will be kept in the server for later to communicate with the application, and the application will also get the id of the server to communicate with the server in the future.

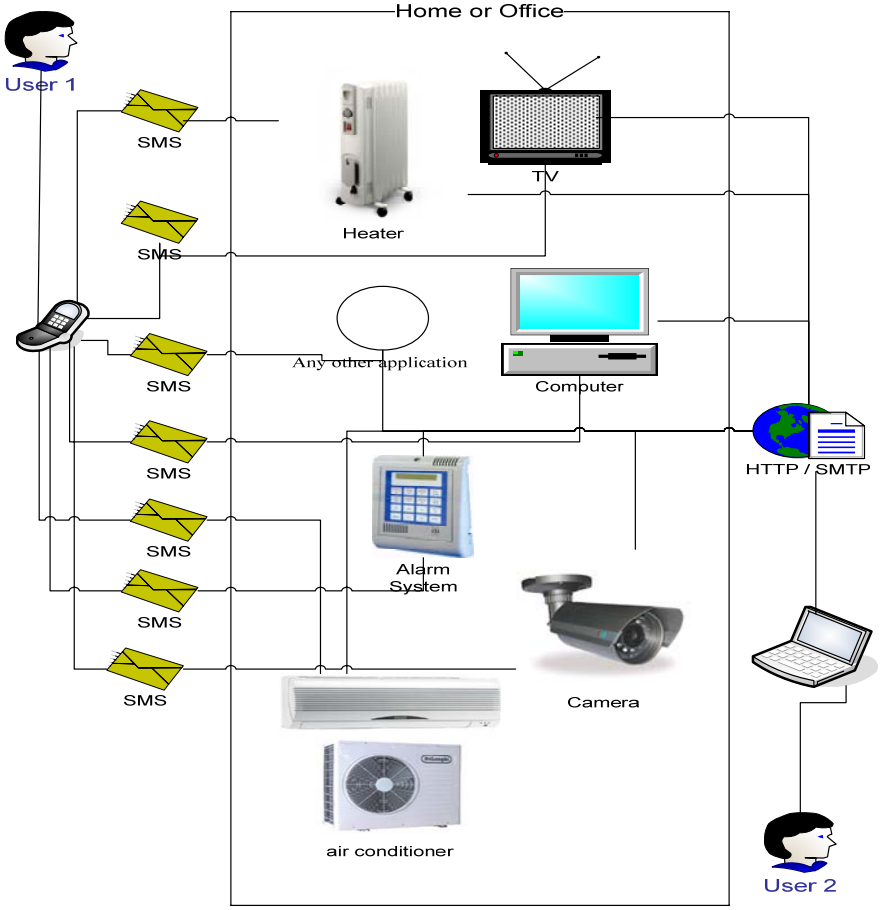


Figure 2: System Two

Figure 2: Shows the second option to implement the system. This option is expensive as each device must have an individual phone number as well as individual IP address. This is easy to build and maintain as they are not integrated with each other. My aim is to build a system like this but also have in mind that our aim is to achieve what's in figure 1.

Technologies:

There are lots of technologies that are out there for mobile devices but the problem is to find the correct option for out project. Each technology has its own advantage and disadvantage and will try and explain them in this part.

There are three main technologies that are out there that can be used to build mobile application:

1) J2ME (java Micro Edition)
2) .NET Compact Framework
3) Flash Lite

Comparison between .NET CF and J2ME (CDC and CLDC)

	.Net Compact Framework	**J2ME Connected Device Configuration**	**J2ME Connected Limited Device Configuration**
Portable code	Intermediate Language (IL)	Machine-independent byte code	
Security	Good	Good	
Portability	No	Yes	
Device requirement	Powerful	Powerful	Pervasive
Cost	High	High	Medium
Market focus	Enterprise	Enterprise	Consumer and enterprise
Platforms	Pocket PC, Windows CE	Major mobile platforms except Palm OS	All mobile platforms
Security model	Simplified .Net model	Full Java security manager	Limited Java 2 model supplemented by OTA

			specification
User interface	Rich subset of Windows Forms	Rich subset of AWT (Abstract Windowing Toolkit), vendor-specific UI libraries	MIDP liquid crystal display UI, PDA Profile subset of AWT, vendor-specific UI libraries
Web services tools	Integrated with VS.Net	kSOAP plug-ins for leading IDEs	kSOAP plug-ins for leading IDEs
Email and PIM (personal information manager)	P/Invoke Outlook APIs	JavaPhone and third-party APIs	Upcoming PDA Profile and third party
SMS	P/Invoke device SMS stack	Wireless Messaging API	Wireless Messaging API
Instant messenger	P/Invoke MSN (Microsoft Network) and other IM client APIs	Third-party APIs for most IM clients including Jabber and Jxta	Third-party APIs for most IM clients including Jabber and Jxta
Enterprise messaging	P/Invoke MSMQ	Proprietary Java Message Service APIs	Java Message Service via third-party toolkits (e.g., WebSphere MQ Everyplace, iBus Mobile)
Cryptography	Third-party APIs	Java Cryptography Extension and third-party libraries	Third-party libraries
Location API	APIs provided by carriers	Third party	Third party

Figure 2 [4]

Both the J2ME and .NET CF are excellent platforms for developing mobile applications. However, each platform has its own strengths and weaknesses. J2ME beats .NET CF to the punch when it comes to portability across a variety of devices, and support for mobile databases and server products. J2ME vendors also offer a wider selection of add-ons and development tools. It is the best platform for mobile solutions on **low-end devices**, geared towards homogenous environments. However, developing and deploying applications on the J2ME platform can be both challenging and tedious. The .NET CF is simple **and ensures lower development costs**. It is also well integrated within Microsoft's solutions and functionality. It is by far the platform of **choice for developing high-end PDAs** running on Windows-only platforms. [3]

Our plan is to make application for low-end as well as high-end devices and the best option for this is J2ME.

Comparison between FlashLite and J2ME:

The big advantage of Flash Lite is that in most cases it has the shorter development cycle compare to J2ME. The advantage with J2ME is that you can do lot more than what you can do with Flash Lite and it's widely supported. This is related to the level at which they work. J2ME is lower-level than the Flash Lite libraries which requires to write more code but provide the flexibility to do more.

For out purpose J2ME is the best option as it's widely supported and provides more functionalities than Flash Lite.

What a developer with experience in Flash Lite, J2ME, .NET CF, and Python has to say:

Short Development cycle: Flash Lite
Medium Development cycle: .NET CF (lots more features)
Long Development cycle: J2ME and C++ (the most features)

It makes sense that J2ME and C++ are the longest development cycles. They support the largest number of handsets and features. .NET CF is not bad to develop for, although it is only for Windows Mobile devices.

With that said, Flash Lite has some advantages:

1. quick prototyping (an emulator would close the loop here)
2. cross platform

http://tech.groups.yahoo.com/group/FlashLite/message/103

Short Message Services (SMS):

The SMS is a service that enables phones to send and receive short text messages on their mobile phones. The messages can be received at any time even during data or voice transmission. The GSM SMS provides a **connectionless** transfer of message with low capacity and **low-performance**. Each message can contain up to 160 characters of Latin alphabets, or 70 characters of non-Latin alphabets such as Arabic and Chinese. To allow messages longer than 160 characters, SMS concatenation and SMS compression are defined in the GSM standards. [6]

Concatenation

Concatenation is the process of concatenating several individual standard short messages into a group to form a long message. Up to 255 short messages can be concatenated. When concatenation is used, each short message contains additional information such as Data Header Indicator. It is used to indicate the use of concatenation, the number and order of the set of concatenated messages. This enables the receiving application to correctly reassemble the concatenated short messages. If any messages in the concatenated sequence are lost, there is no automatic retransmission mechanism. Concatenation can be used for transferring messages contain more characters than standard. However, it is impractical for high volume information transfer. [6]

Compression

With compression, the extra characters can be compressed into a single short message.

The choice of a compression algorithm is limited by the availability of the J2ME compression implementation. Although the compression API is bundled with J2ME Connected Device Configuration (CDC), which is targeted for devices with at least 2 MB of memory, most current cell phone devices support only Connected Limited Device Configuration (CLDC) and the MIDP standard built on top it. The minimum memory requirement for CLDC is 128 KB. Therefore, developers must find third-party libraries or write their own. I found three:

- JCraft's compression library uses the zlib compression algorithm. It is open source, well documented, and has a large following; however, the library's size is a bit large.
- Java4Ever has a gzip (GNU zip) implementation (3.27 KB). It is under a LGPL license and is also open source.

- I like TinyLine's GZIPInputStream (5.32 KB) because it extends `java.io.InputStream` and follows the same Decorator pattern as other Java stream classes. It supports `skip()`, `mark()`, and other basic I/O functions. The tool's author is very responsive to questions and the library is straightforward to use:

[7]

Binary messaging

The SMS can be configured in character or binary mode. Binary messaging enables more efficient transfer of data. [6]

Encryption and security

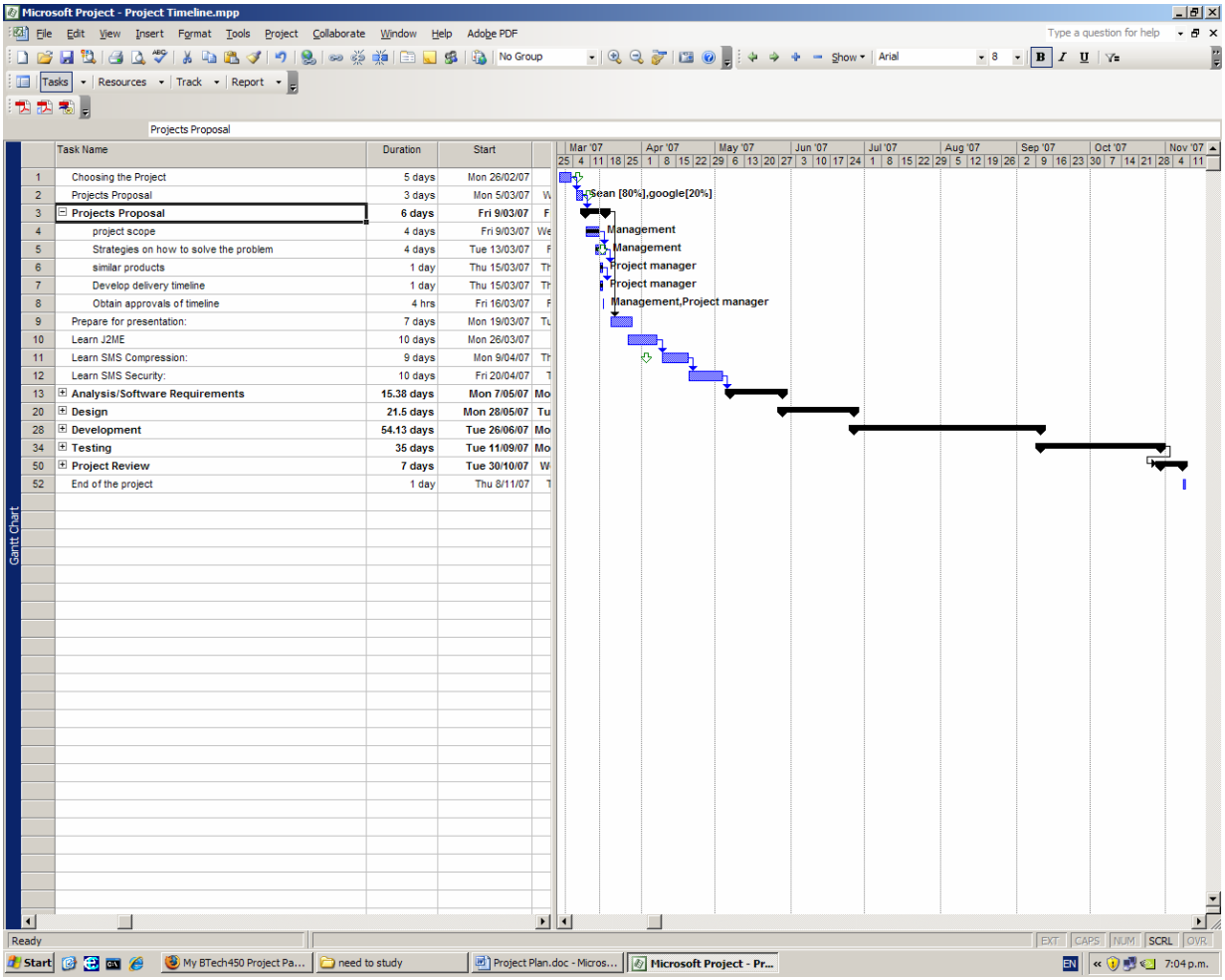
To ensure that short messages are not corrupted or intercepted, data integrity is incorporated into the GSM standard. Short messages are routinely encrypted over the radio path between the MS and BSS using the IA5 encryption algorithm. IA5 is sufficient for most routine short messaging requirements. For optimal security in applications such as mobile banking, end-to-end encryption is advisable between the sending and receiving Short Message Entities. [6]

Different alphabets support

There are two different GSM SMS alphabets: UCS2 and the GSM 7-bit Default Alphabet. UCS2 (Universal Multiple Octet Coded Character Set 2) is derived from the ISO standard ISO/IEC10646-1. UCS2 incorporates all major languages around the world. The 7-bit Default Alphabet is derived from the ASCII character set. Each Latin character uses 7 bits to code, whereas the non-Latin characters require 16 bits. This results each non-Latin short message has a maximum length of 70 characters and such applications may therefore need to use compression or concatenation for greater information transfer. In addition to UCS2 and the default 7-bit alphabet, GSM specifies also eight bit (binary) data. [6]

There is no standard protocol for SMS transmissions. The Short Message Peer to Peer Protocol (SMPP) is most commonly employed, followed by the Universal Computer Protocol/External Machine Interface (UCP/EMI).

Expected Time Line:



Reference:

1) URL: Name: **http://www.1touch.co.nz/?p=smart**

2) URL: http://www.nailed.co.nz/Home_Technology_611.aspx

3) URL: http://www.tometasoftware.com/J2Me_vs_Net_whitepaper.asp

4) URL: http://www.must.edu.my/~dwong/resources/mobile_commerce_web/j2mevsnetcf.html

5) URL: www.tml.tkk.fi/Opinnot/T-109.551/2003/kalvot/Mobile_Entertainment.doc

6) URL: http://www.javaworld.com/javaworld/jw-05-2005/jw-0502-midp.html