

Distributed Polling System Mobile Communication

Version 1.0

Table of Contents

1. Introduction	2
1.1. Purpose of this document	2
1.2. Document organization	2
1.3. Intended Audience.	2
1.4. Scope	2
1.5. Definitions and acronyms	2
1.5.1. Acronyms and abbreviations	2
1.6. References	3
2. Requirements for mobile communication	3
3. Basic Concepts	3
3.1. SMS and SMSC	3
4. Solution Alternatives	4
4.1. Establish connection with SMSC or SMS gateway of a Telecom operator.	4
4.2. Establish connection with SMS Resellers, SMS Brokers or SMS Service Provider.	4
4.3. Send an Email to the Mail Domain provided by the Telecom Operator.	5
4.4. Connect a mobile phone to a computer.	5
4.5. Decision	6
5. Communication Mechanism	6
5.1. Application Overview.	6
5.2. Implementation Overview.	7
5.3. DPS Implementation of Sending SMS.	8
5.4. Ensuring SMS Delivery and Handling Lost Messages	10
5.4.1. Services need to be configured from DPS Application	10
5.4.2. Services provided by mobile device or SMSC	10
6. Future Enhancements	11

1. Introduction

1.1 Purpose of this document

This document provides technical details for mobile communication sub-layer which is a part of DPS Communication layer. This document also proposes design and implementation mechanism for extending the application to use other modes of communication (as mentioned in the SCORE project document).

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*, describes contents and scope of this document
- Section 2, *Requirements for mobile communication*, presents the needs for mobile communication in the overall DPS application.
- Section 3, *Solution Alternatives*, In order to satisfy needs explained in Section 2 what are the possible solution available, which one is selected and Why.
- Section 4, *Basic Concepts*, Describe the terminology and concept for understanding the working of selected design alternative, both from telecommunication domain and technology perspective.
- Section 5, *Communication mechanism*, describes how the DPS System implements the Mobile Communication.
- Section 6, *Future Enhancements*, This section explains how to extend the DPS to support other means of mobile communication, i.e. voting through thick client on the mobile device and voting through Voice mail apart from SMS.

1.3 Intended Audience

Intended Audience of this document is:

- Customers (Score supervisor and evaluators)
- Course and Project Supervisors (Prof Ivica Crnkovic and Senior. Lecturer Rikard Land)
- Project Team
- Future users and developers of DPS application.

1.4 Scope

This document is an attempt to cover all aspects of Mobile communication for DPS from Requirements to technical details to possible future extensions to support other mechanism of communication.

1.5 Definitions and acronyms

1.5.1 Acronyms and abbreviations

Acronym or Abbreviation	Definitions
DPS	Distributed polling system
SMS	Short Message Service
SMSC	Short Message Service Center
MSP	Mobile Service Provider
SMSG	Short Message Service Gateway
SIM	Subscriber Identity Module

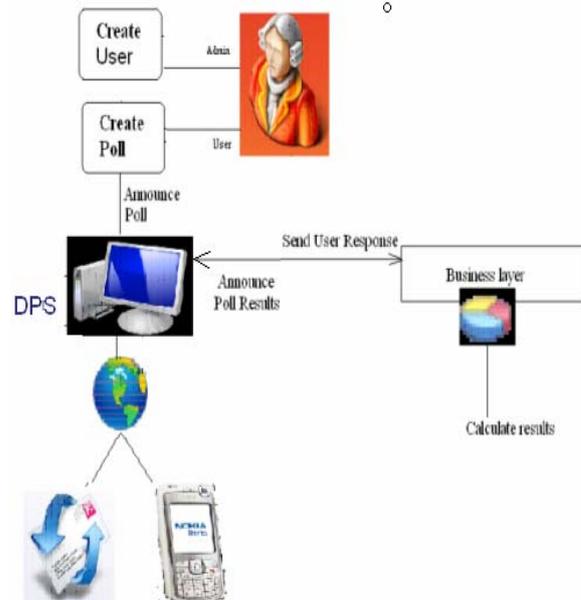
1.6 References

No.	Name	Description
1	DPS_Requirements_Specificati on document	Requirements specification for DPS system
2	DPS_Design_Specification document	Architecture and Design details of DPS software System

2. Requirements for mobile communication

DPS is a voting system which facilitates users to vote on a certain issue virtually present anywhere in the world connected by an electronic means of communication (mobile, email etc). One major means of communication can be through mobiles ranging from SMS, voice based to Image or video options. We restricted our scope to SMS but provide mechanism for future extension to other mechanisms.

- A user needs to be intimated by SMS that he/ she has to cast a vote.
- User needs to be provided with the details of voting, from voting subject to facts and figures for making a decision.
- A user can cast his vote by sending SMS. Hence receiving, validating and processing of his vote is required.
- The results can be announced and broadcasted to them by SMS.
- Users can be allowed to have a discussion facilitated by DPS system using the mobile communication mechanism.



3. Basic Concepts

3.1 SMS and SMSC

Short Message Service (SMS) is a technology that enable sending and receiving of messages between mobiles. The message size is fairly small 160 characters for 7 bit encoding or 70 char for 8 bit encoding (support other language such as Latin, Chinese characters through 8 bits). SMS is supported by all GSM mobiles and also support binary data apart from Text, to transfer ring-tones picture, logos, and wallpapers and so on. To overcome this size limitation Concatenated or Long SMS are used. In long SMS, bigger message is broke into smaller parts and each one is sent as separate SMS. At the reception end these parts are combined and presented to user. Sending pictures, animation and voice is difficult due to size limitation thus Enhanced Messaging Service (EMS) is introduced but is rarely used.

When an SMS leaves the mobiles phone its operations are handled by SMS Center (SMSC). The functioning of SMSC can be comparable to what Router does in Internet. When message is received by SMSC, its task is to forward to the recipient or other appropriate SMSC or SMS Gateway of other network. Another major function of SMSC is to temporarily store the SMS when recipient is not available or his mobile is switched off. When the user becomes available the stored SMS is forwarded.

4. Solution Alternatives

We researched on finding the ways in which a computer program can send a message to mobile. As none of the team member was earlier familiar with Mobile communication technology, we searched Internet, books, tutorials and few research papers. Our found following four ways:

Establish connection with SMS center (SMSC) or SMS gateway of a Telecommunication operator.

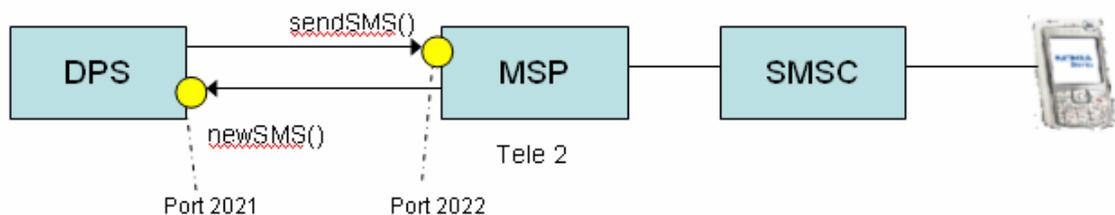
Establish connection with SMS Resellers, SMS Brokers or SMS service provider.

Send an Email to the Mail Domain provided by the Telecommunication operator.

Connect mobile phone or GSM/GPRS modem to a computer.

4.1 Establish connection with SMSC or SMS gateway of a Telecom operator.

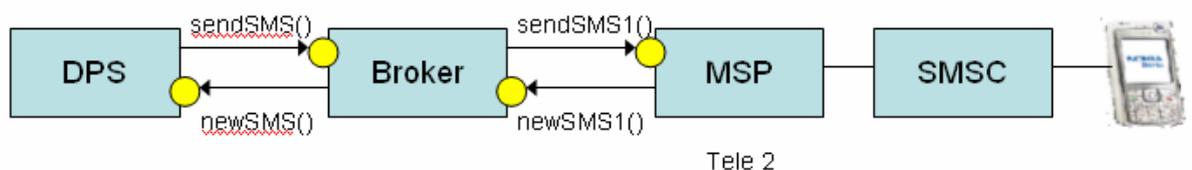
Usually Mobile Service Providers or Telecom Service Providers establish servers on the Internet, through which computer programs can establish connection with them and send the data over this connection. The data will be forwarded by these servers as SMS to a suitable SMSC and finally will be delivered to the users of the mobile phones. For communicating with these servers computer program need to follow the communication protocol establish by the MSP. We first tried this approach and try contacting 2 telecommunication operators “Telia” and “Tele2”. After large number of Emails and telephone calls Tele2 has forwarded a 500 plus page technical document. Unfortunately this document is in Swedish language and even after language conversion many of the terms was not understood by us (as none of us are Swedish native). For this service MSP charge a huge amount of money and is suitable when large number of SMS has to be sent.



Considering the cost factor and unavailability of technical details for communication protocol we dropped this approach.

4.2 Establish connection with SMS Resellers, SMS Brokers or SMS Service Provider.

For users that do not have requirements for sending bulk SMS this is one more option. They can contact SMS service providers or brokers that sell these services. These brokers in turn have connection with telecom operator as explained in section 4.1. These brokers sell the services to a number of users. Thus the cost that each user pays is pretty less and large number of users share this cost with broker in between.



This is similar to taking a high bandwidth connection of internet from Internet service provider and redistributing bandwidth it to other internet users.

We contacted broker companies, in 1 such case it is required to install tool on our end

provided by this company. When asked for evaluation version, the organization does not provide evaluation version to students, it only provides to corporate clients. Another organization does not require such installations but their fee is pretty high. We not only need to subscribe to this service not only till development of the application but till the score evaluators evaluate our final product. Moreover these companies are usually suitable for advertisement SMS but not suitable for sharing SMS that contains internal organization decisions/details. For voting purpose where business critical information may be shared with the key employees of the organization this approach might be risky. Considering the above limitations this approach is also dropped.

4.3 Send an Email to the Mail Domain provided by the Telecom Operator.

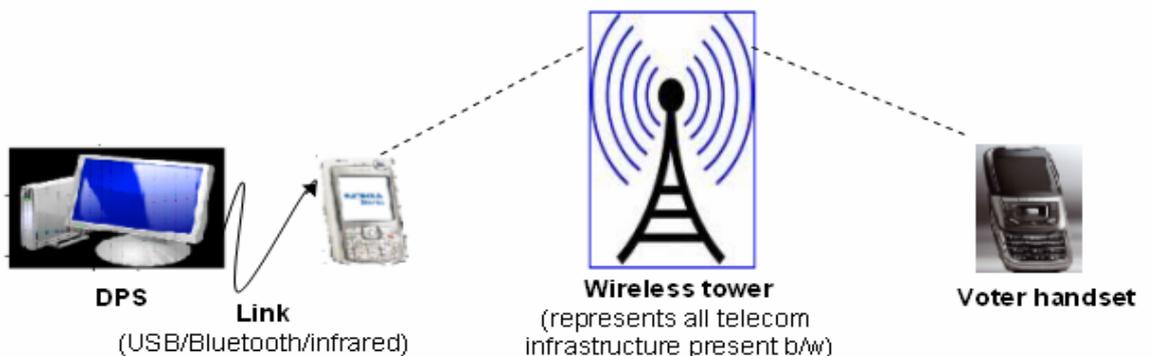
Some telecom operators provide an Email domain and corresponding to each mobile subscriber they provide an email ID. If we forward the message to this email ID, the corresponding SMS will be sent to the subscriber. For ex: Idea Cellular in India, Tele 2 in Sweden also provides this service. To send an SMS to mobile no 0046765836758 send an email to 0046765836758@tele2.se. Tele 2 will forward the message to the mobile. As we are also implementing voting through Email service, using the email mechanism to send sms might solve the problem.



The Disadvantage of this approach are not all telecom operator provides this service. If a voter which is a subscriber of the telecom operator, which does not provide this service, the mechanism fails. Moreover the Email server of the telecom operator does not provide an acknowledgement that the message has been received by the subscriber. Due to these limitations this approach is also dropped.

4.4 Connect a mobile phone to a computer.

One more solution is to physically connect a mobile to the computer and let the computer program control the operations of the mobile device. This approach is similar to the way a user sends an SMS by hand-typing the text and pressing the buttons, instead the computer program does the same instead of a human. The connection between mobile and computer can be through USB cable, Bluetooth or an infrared link. In place of a mobile a GSM/GPRS modem can also be used.



This approach has many advantages. For sending SMS this approach only requires a mobile

phone with a valid SIM (Subscriber identity Module) card to be connected to the computer. The data safety, encryption and communication protocol all these details are already implemented by service provider and handset manufacturer. We don't have to bother about these issues. The cost of sending SMS is same as the cost of sending SMS by a person through his handset and depends upon the type of connection the subscriber has taken. For us we used tele2 SIM card which allow sending unlimited SMS in very low fix monthly charges. This approach is easily extensible to other mechanism such as voice mail and picture transfer, as explained in section 6 of this document.

Disadvantage are 1) Speed: is that this mechanism is pretty slow and can send only a limited number of SMS per minutes. Considering the fact that only a small group of people are using the DPS application to take a business decision the impact is less. Moreover a mobile phone can be replaced by a GSM/GPRS modem, with no major changes, to overcome the speed disadvantage. We used 2 mobile phones instead one for sending and other for receiving to increase the speed. According to analysis of one mobile expert, the sending message in text mode and receiving in binary mode is fast. So we implemented the same way.

2) Portability: All mobile do not support all AT commands. There is no universal standard that all mobile manufacturers should support specific AT commands and is fully depends on the manufacturers decision. Our experience is that even different handset models from the same manufacturer support different command set. Thus our DPS application is configured to work with Nokia N95 for sending SMS in text mode and Sony Ericsson z550i for receiving SMS in binary mode. For working with other model the program might need to be configured. This is one of our main concerns as SCORE evaluators need to connect mobile to the computer and they might not have the same mobiles as we have used. In the *DPS_Installation_Manual* document we describe how to test if a mobile supports the required AT command or not.

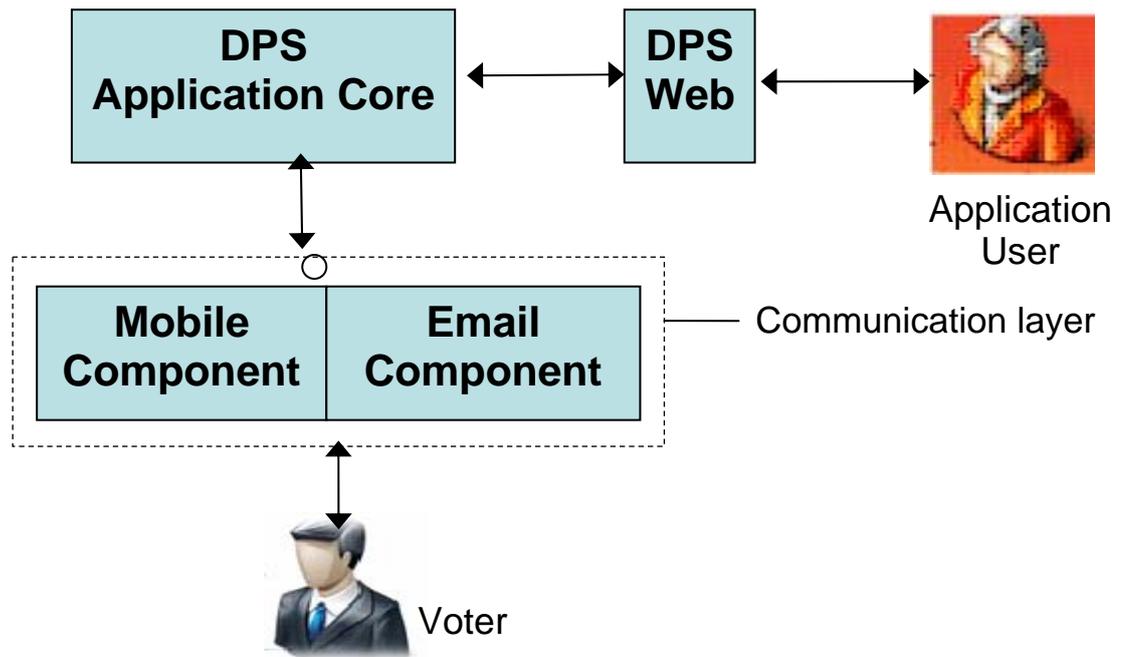
4.5 Decision

Considering the pros and cons of each approach in Section 4.1-4.4 the approach is implemented by DPS application for mobile communication is "Connect a mobile phone to a computer". The main reason of choosing this approach is Cost and Time. With no extra cost then sending the SMS, this approach also enable us not to depend on telecom operator or SMS brokers and spend time in communicating and searching for them

5. Communication Mechanism

5.1 Application Overview

The following diagram depicts the position of the Communication layer in the overall DPS System.



Communication layer and its Component (sub layer) within DPS System

The Communication layer is responsible for handling all communication coming in and going out from the DPS, to the voters. For ex: When a call for voting is to be made all the information for, a call for voting is provided by application core or business logic to the communication layer. Now it's the comm. layer which handles the announcement to Voters mobile or email.

5.2 Implementation Overview

For sending messages to users we adopted the approach of connecting the mobile handset to computer, and then let computer program control the functioning of the mobile. The mobile must have a valid SIM as well as should have sufficient balance to send SMS. As computer have Instruction Set, mobiles also have also similar mechanism (AT commands) but on a small level and with limited functionality. Even the software's installed on the mobile ultimately uses this representation of mobile instruction set (AT command) to accomplish its low level functionality. We can use this AT commands to send and receive SMS and do certain basic SMS operation to ensure the successful delivery of the message.

Suppose we want to know the manufacturer of the mobile device connected to our PC. We send the command "AT+CGMI" to the connected mobile, the result would be the Manufacturer Identification.

Few AT Commands to send SMS

Command	Description
AT	To test the connection, just like we have ping command to test network
AT + CMGF	Set Working Mode to Text Mode
AT + CMGW	Set the Recipients mobile number and message text into mobile memory
AT + CMSS	Send the SMS stored in memory to be delivered

Another issue for sending SMS is how to get the SMSC address of the telecom operator. The address of SMSC is embedded within the SIM card and is fixed. This address has the same format as an international mobile number has. For a computer program to get SMSC address we use following AT command.

AT+CSCA	Command name in text Service Center Address or SMSC address
---------	---

Now the question arises of how to send these AT commands from computer to mobile which might be connected through USB / Bluetooth or Infrared. One solution is to use a Terminal Program. A terminal program sends the typed characters to the mobile and in turn collects the response sent by mobile phone and displays them on the screen. One such terminal program by Windows XP or lower releases is HyperTerminal.

Another solution is to use the Java Serial Communication Application Program Interface (Comm. API, javax.comm) to communicate with any device attached to the serial port. There are two ways in which the communication between program and mobile device can occur a) Synchronous b) Asynchronous

Synchronous: When program polls the serial port for data, it will wait till the data becomes available (or until a timeout occurs).

Asynchronous: The program does not wait for data to become available.

In our Implementation we used Asynchronous mechanism (Please refer to code below).

5.3 DPS Implementation of Sending SMS

Following is the simplified code for sending SMS we used in DPS (low level details are removed).

Modified SMSSender.java:

```
public class SMSSender implements Runnable{

    public final static boolean IS_SYNCHRONOUS=false;
    public int status=-1;
    private Thread currentThread=null;
    private String recipient=null;
    private String smsMessage=null;

    public int sendSmsMessage (String recipient, String smsMessage){
        this.recipient=recipient;
        this.smsMessage=smsMessage;
        currentThread = new Thread(this);
        currentThread.start();
        return status;
    }

    public void run(){
        SMSConfigurator aSMSConfigurator = new
            SMSConfigurator(recipient,smsMessage);
        try{ aSMSConfigurator.send (); // Sends SMS

            //if SYNCHRONOUS wait until success
            if (IS_SYNCHRONOUS == true) {
                while (aSMSConfigurator.status == -1){
                    currentThread.sleep (2000);
                }
            }
        }catch (Exception e){
            //<< Error handling and reporting Back to Communication Layer >>
        }

        this.status=aSMSConfigurator.status ;
        aSMSConfigurator=null;
    }
}
```

Modified SMSConfigurator.java:

```

Public class SMSConfigurator implements Runnable {

    private String portName="COM2"; //The name of the port.
    public final static int baudRate=9600; //The baud rate.
    public final static int flowControlIn=0; //Type flow control, receiving.
    public final static int flowControlOut=0; //Type flow control, sending.
    public final static int databits=8; //The number of data bits.
    public final static int stopbits=1; //The number of stop bits.
    public final static int parity=1; //The type of parity.

    private SerialParameters defaultParameters= new SerialParameters
(portName, baudRate, flowControlIn, flowControlOut, databits, stopbits,
parity);

        public int status = -1;
        SerialConnection mySerial =null;
        Thread aThread=null;
        String recipient=null;
        String message=null;
        private String csca=null; // the SMSC or Message Center address

    public SMSConfigurator(String recipient, String message){
        this.recipient=recipient;
        this.message=message;
    }

    // Establish a Serial Connection and Start the Thread
    public int send () throws Exception{

        SerialParameters params = defaultParameters;
        mySerial =new SerialConnection (params);
        mySerial.openConnection();

        aThread=new Thread(this);
        aThread.start() ;

        return status;
    }

    // This method send SMS using AT commands
    // The below method in real quite complicated
    // We here present a simplified and reduced version of it
    // All Error handling is removed from it.
    // Following steps are like Algorithm and is NOT EXECUTABLE.

    public void run(){
        // Step 1: Check if the device is connected and is working
        // Expected Output=OK signal
        executeATCommand("AT");

        // Step 2: Get the SMSC address from Mobile SIM Card
        // Expected Output=Address of SMSC in International mob number format
        csca = executeATCommand("AT+CSCA");

        // Step 3: Set the device to work in Text Mode
        // Expected Output=OK Signal
        // If ERROR signal means mobile does not support Text model
    }
}

```

```

Status=executeATCommand("AT+CMGF=1");

// Step 4: Set the address of SMSC
// Expected Output=OK Signal
Status=executeATCommand("AT+CMGW="+csca);

// Step 5: Send the SMS Message to Mobile storage/ memory
// Expected Output= A number which is location of SMS in memory
sendSteramToDevice(message);

// Step 6: Send the SMS Text Message from mobile memory
// Expected Output=OK Signal
executeATCommand("AT+CMSS=1");
}

```

The mechanism for reading incoming messages from voters is similar but is more complicated in term of operation and AT commands.

5.4 Ensuring SMS Delivery and Handling Lost Messages

Error handling are served by two kind of services 1) Services provided by the mobile device which silently handles error without bothering the user or computer application 2) Some services need to be configured by the computer application to have more strengthen message policy

5.4.1 Services need to be configured from DPS Application

- a) **Validity Period:** If a mobile is not available (switch-off) or is unreachable the SMSC stores the message for a certain period of time, if the user will become available during this period the message will be delivered to him. This period is called validity period and is generally between few hours to few days. For DPS it is not desirable for the application to wait for such long time after an intimation of poll has been announced. For ex: within 2 min if the message is not reached we need to remove the message from SMSC and look for other electronic mode of communication such as email. It is possible to set validity period using another AT command.

AT+CSMP=[<fo>[vp][...]], Set text Mode Parameter, where
 <vp> is validity period depends on <fo> which is setting of time in string or integer format

- b) **Message Status Report:** It is possible to have an acknowledgment for whether the text message is received by the mobile user. By default this option has been turned off and no ACK is received. When this option is turned ON from application by setting fields in the SMS, the SMSC will respond back i.e. sends an acknowledgment SMS to the sender (our connected mobile) confirming the status (success or failure) of the sent message.

5.4.2 Services provided by mobile device or SMSC

Following two services are automatically provided by the mobile device and SMSC for dealing with erroneous or lost messages. We need not to handle them in our code but their knowledge may be helpful

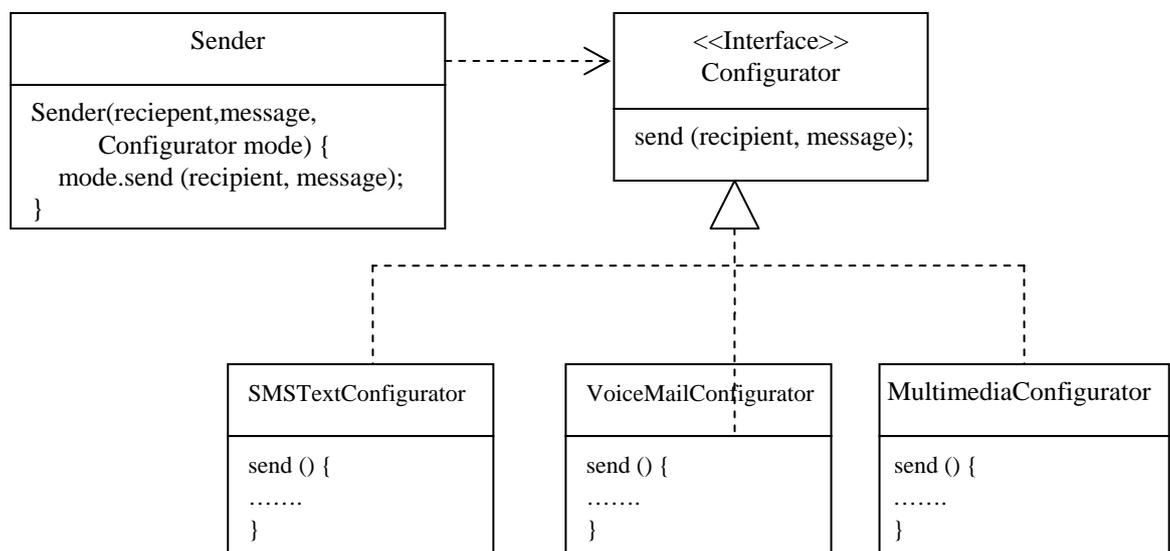
- a) **Message Submission Report (b/w Sender and SMSC):** After an SMS leaves the mobile phone and reach SMSC; the SMSC will send a message submission report to mobile. If the SMSC receive an erroneous or incorrect format SMS etc. the SMSC will send a negative report to mobile. The mobile will again transmit the SMS without the application having knowledge of it. If after certain time period, even if the positive report is not received by the mobile, the mobile consider it as a lost message and transmit the SMS again.
- b) **Message Delivery Report (b/w SMSC and Receiver):** If the receiver mobile receives an erroneous message, it will send a negative Message Delivery Report to SMSC. The SMSC will transmit the message again without the knowledge of Senders mobile or

application. If after certain period of time, if SMSC does not receive either positive or negative Message Delivery Report, it will consider as a lost message and will transmit again.

6. Future Enhancements

Apart from text SMS it is possible to send and receive pictures, multimedia, audio, voice mail. The mechanism of AT command explained in this document for sending text message applies equally well to these advance options of mobile communication. Not all mobile support audio and multimedia transfer, thus it depends upon the capability of the sender's mobile (attached to the computer) and receivers mobile. If a mobile supports sending of audio, it usually supports the corresponding AT commands for sending audio. The only requirement is the send () method of Configurator should be implemented using the corresponding AT command set.

In the following UML diagram, we propose a Design where it is possible to incorporate new mode of communication without changing existing Infrastructure or Code.



Suppose a new medium of communication “Multimedia-Graphs” have to be supported, a new class (name MultimediaGraph) which implements the Configurator interface has to be implemented. Rest all remains the same. The details of how the communication will happen using AT commands will be handled by the send method of this class.

The above design is based on two principles:

- A Interface reference variable can point to an class object of its implementing class: The reference variable “mode” of interface Configurator (in the class Sender) point to either of the object of class SMSTextConfigurator or VoiceMailConfigurator.
- Dynamic Method Dispatch: Call to a method will be decided at runtime and depends on which object the invoking reference variable is pointing. The object “mode” will call the send method, whether the send method of SMSTextConfigurator or send method of VoiceMailConfigurator will be called depends on the object it refers to. Thus it it not require to hardcode whether an SMS is to be sent or voicemail.

The main point here is that different modes of communication can be supported by using different AT commands and rest all remains the same.