DEVELOPMENT OF AN SMS-BASED OUTAGE REPORTING SYSTEM

Kehinde ALOWOLODU
Electrical and Electronics Engineering
The Federal University of Technology,
Akure, Nigeria
alowolodu@futa.edu.ng

OLUFEAGBA Jimi
Electrical Engineering Department
University of Ilorin
Ilorin, Nigeria
lobojay12@yahoo.com

Olufunso ALOWOLODU
Computer Science Department
The Federal University of Technology,
Akure, Nigeria
funsoalowolodu@yahoo.com

ABSTRACT

Traditional outage reporting in Nigeria is done by consumers visiting the service station of the electricity company and making reports of outages. This presents many problems such as inaccurate data management, slow response and uncoordinated outage management among others. Developed outage reporting system are based on fixed land lines and these are not very common in Nigeria. The Global System for Mobile (GSM) has over the past decade covered most of the country. A real-time outage reporting system based on the Short Message Service (SMS) feature of the GSM was developed and implemented. The solution is a GSM modem interfaced to a computer which would receive customer originated messages, process them and update a database and thus alert essential units in the organization for action in real time while providing the means of relatively accurate collection of data useful for planning and other functions. The software was developed with Microsoft C# and MapWindows which provided a means of overlaying the faults reported on a digital map of the coverage area. The system proved to provide a shorter mean time to repair of faults on the Akure distribution area.

INTRODUCTION

The purpose of fault management is to detect, isolate, diagnose and correct the faults during network operations while causing as little disruption as possible to the consumer. Therefore the primary function of fault management is to sustain normal operations and ensure the highest level of network reliability and availability. In this sense, fault management serves as the foundation of other network management functions [1].

In most other places, outages at the low-voltage level are not captured by sensors in the system but are reported by the consumers. At this level of the network the customers are the eyes and ears of the utilities. This puts a lot of stress on the staff in the operations and maintenance department since they have to manually organize customer's complaints which may run into thousands per day. An added feature is that the same fault/outage might have been repeated by same consumer several times that day.

CUSTOMER PREMISES TO NETWORK LINK DATA

Consumer Premises to Network Link is a set of data that enables each of a utility's customers to be uniquely identifiable from the perspective of the electrical network. This is done by associating the consumers' premises to the best defined network component. A primitive system would only be able to identify customers as far as identifying the relevant supply transformer. Clearly this will give room to a lot of ambiguity in cases where transformers are large and many consumers are connected to one. A better system would associate consumers to poles from which their service wires are connected in most instances faults reported by one consumer defines the state of all others fed through the same component.

Once the Premises have been linked to a network device, it becomes possible to relate a call from the customer to known incidents on the network, or to infer incidents on the network based on information from customers. This link is also critical in being able to accurately measure the quality of supply delivered to a customer

REVIEW OF SMS TECHNOLOGY

In traditional outage management system, outage reporting is usually by voice telephone calls. The consumer is usually identified through his/her telephone number as registered in the consumer information system or the billing system. Most consumers in Nigeria do not have wired telephone lines although GSM have proliferated the society. An important part of the GSM specification is the SMS which is supported by virtually all mobile operators.

The GSM SMS provides a means of providing alphanumeric (letters and numbers) or other kind of information to the user of a mobile telephone. GSM distinguishes two different kinds of messaging services [2,3]:

a) dedicated service between two parties which requires the establishment of a dedicated point-to-point link. thus the name point-to-point SMS: and
b) broadcast service between the network (through one or more base stations) and all users within a cell or service area. This is called point-to-omnipoint or cell broadcast SMS.

SYSTEM DESCRIPTION

Figure 1 shows the block diagram of the system. The system consist of the SMS reception and processing modules, modules for updating databases and alerting managers and a module to update map of the distribution areas.

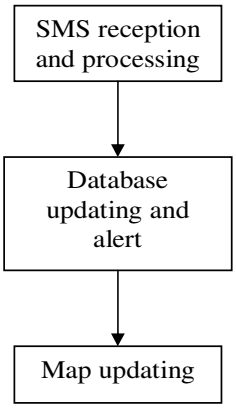

Figure 1. System Block Diagram

The developed system is as shown in Figure 1.

DATABASE DESIGN

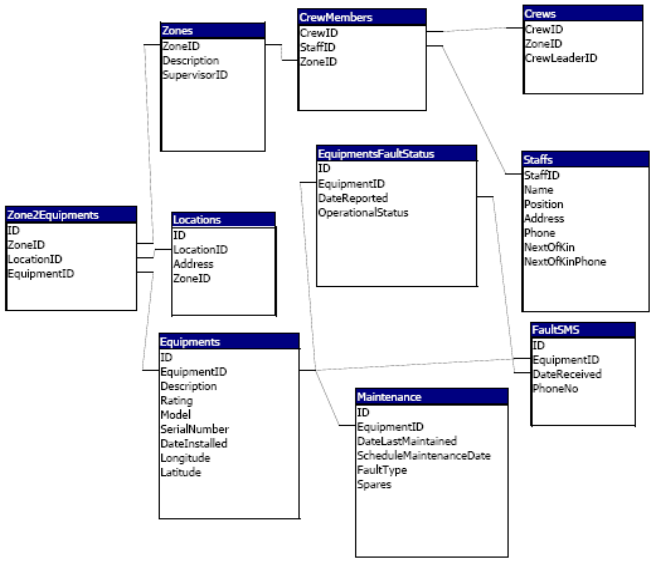

Figure 2. Relationships in database for outage reporting system

In order to report and manage outages in a distribution system, a database with the structure and relationships in Figure 2 was created. One of the hard decisions that must be made is choosing between a large number of database tables and rigorous coding. Large table tend to slow down applications as large memory space must be committed in order to read them into memory. This will become more pronounced in a network application where the tables are to be downloaded from the database server into client systems over the network. In this application, smaller tables are used.

Since the system must be backward compatible with existing systems, the database for equipment status can be manually updated so that customers can make face to face reports at the fault centers and repair crews can also fill in paper reports of repairs made.

PORT INITIALIZATION

The Microsoft C# and all other programming languages handle the GSM phone via a serial communication interface. Older GSM phones have to be interfaced through special modules that convert the native CMOS logic of the phone to TTL logic of the serial communication port of the computer. The introduction of the USB interface to computers eliminated the need for logic translation so that phones can be connected directly to computers via USB port.

Computer to phone communication through the USB port requires an appropriate driver usually specific to the phone type and model.

Communication with a serial device begins with the port initialization. This is done by instantiating a new serial port object, binding it to the serial port to which the modem is connected and opening it for reading and/or writing. The baud rate, encoding, the number of data bits, the parity bit and the name to be associated with the port are important port parameters to be set at initialization.

A problem presented by this traditional approach is that one must know the port number to which the mobile terminal is attached, which is straightforward in traditional COM ports which are pre-numbered. In the case of USB devices, the port number is not static even with the same USB port. Port numbers are assigned on as-available basis whereby a port assigned port number COM5 can be reassigned as COM7 when next the device is attached.
In order to eliminate this constraint, a specific port number is uniquely assigned to the USB port so that whenever the phone is connected, the port number is automatically set.
Another alternative approach to eliminate this constraint is to write a procedure that cycles through the COM ports and determine whether a modem is connected. It then automatically assigns the port number.

RECEPTION AND PROCESSING OF SMS

Communication with the mobile terminal is done using a special command set known as AT (ATtention) commands. This is the same set of command used with old modems, but with an extension for GSM-specific services. A major service in the GSM specification is the short message service (SMS) class of service. Working with SMS in programming languages is not straight forward with the AT command. SMS are stored in memory and are referenced only by the memory location in which it is stored. This is not the way it is displayed on the GSM hand set where SMS

are sorted by delivery date, phone number or name of sender. Messages are in four categories:

a) Received Read: these are messages that were received from another device and have been read by the user. It will contain the address of the sender (phone number), the timestamp (date and time when the message was received at the message center);
b) Received Unread: these are messages that were received from another device but have not been read. It contains the same information as the received read, but the read flag have been set. This can be unset manually in some mobile terminals;
c) Store Unsent: these are messages originating from the mobile terminal but have not yet been sent. It will usually contain the address of the intended receiver, validity information, handshake options and the message; and
d) Store Sent: these are messages originating from the mobile terminal and have been sent. It will contain the information in the store unsent and the timestamp of the time when the message center acknowledges the receipt of the message.

Only the received messages are of interest in this paper, these are usually stored in the inbox of the SMS folder of the mobile terminal.
Texts are communicated to the computer in one of two modes:

a) PDU, this is as defined in the GSM 03.40 and GSM 03.41 standards. This is usually the default mode when a mobile terminal is connected to a computer. It is not a human readable format;
b) text mode which is human readable.

In order to change the text mode, the AT+CMGF command is issued, the syntax of this command is
AT+CMGF=mode, where mode is either 0 or 1.
Mode=0 stands for PDU format while mode=l changes the interface mode to normal human readable text format.

In order to read the messages, the memory index is retrieved by listing all the messages.
This is accomplished with the AT+CMGL command.
The syntax of the command is AT+CMGL=status- where status is one of the following:

PDU	Text	Description
0	"REC UNREAD"	received unread message (i.e. new message) (default)
1	"REC READ"	received read message
2	"STO UNSENT"	stored unsent message (only applicable to SMs)
3	"STO SENT"	stored sent message (only applicable to SMs)
4	"ALL"	all messages (only applicable to +CMGL command)

Each message to be listed is represented in the format:
+CMGL: <index>.<stal>,<oa/da>,,[,<tooa/toda>.<length>]
<CR><LF> <data>
where
<index> - message position in the storage
<stat> - message status
<oa/da> - originator/destination number
<tooa/toda> - type of number <oa/da>
<length> - text length
<data> - TP-User-Data
To extract the index, every word between a +CMGL: and a comma (,) is extracted and appended to a list. These are the indexes of SMS in the phone memory.

Messages are extracted with the AT+CMGR=index command, where index is one of the indexes read into a table. A message read and extracted into a database is then deleted with the AT+CMGD=index command.
It would have been easier if new messages are indicated to the computer, but not all mobile terminals support the new message indication which can be turned on so that when a new message is received, the memory index of the new message is communicated to the computer, so that it can be read and the message left in the memory. An apparent shortcoming of this approach is that in a situation where by thousands of messages are received, daily or weekly, the phone memory will be full quickly and new messages cannot be received. Consequently messages received and successfully read need to be deleted from the phone memory.
An abstract data type was created for the SMS and this class was implemented by the class definition listed below.

```
public class ShortMessage
    {
      private int index;
      private string status;
      private string sender;
      private string alphabet;
      private string sent;
      private string message;
      private string dates;
       public int Index
       {
           get { return index;}
           set { index = value;}
        }
        public string Status
       {
           get { return status;}
           set { status = value;}
        }
        public string Sender
        {
           get { return sender;}
           set { sender = value;}
        }
        public string Alphabet
        {
           get { return alphabet;}
           set { alphabet = value;}
        }
        public string Sent
        {
           get { return sent;}
           set { sent = value;}
        }
        public string Message
        {
           get { return message;}
```

```
            set { message = value;}
        }
        public string Dates
        {
            get { return dates; }
            set { dates = value; }
        }
    }
```

This class definition allows a short message to be broken into components that represents (from the program perspective) the data contained in an SMS.
The collection of short messages read is stored in a list which is then recorded in the SMS table. This table is used to update the status of equipment as necessary.
The format of the text message to report fault is FAULT network equipment ID (for example to report fault from a consumer connected to transformer T0020. (the message FAULT T0020 is sent to the service number.) Messages that indicate the same fault is classified as a single fault even when there are multiple reports.
SMSs from the field crew to indicate that a fault has been rectified is of the format (REPAIR <EQUIPMNTID>),.
SMSs that do not start with the keyword (either in lowercase or uppercase) is not entered as a message to report fault.

INTERFACE DESIGN
Although many interfaces are implemented the most important interface is the indication of where faults are reported. Two such interfaces are implemented: in table format (Figure 3) and on a map format (Figure 4)

The spatial database was designed such that the primary key in the equipment table was the ID assigned to the network link equipment (for example the first transformer is T0001, second transformer is T0002 etc.) The reports are stored in a table and the faults are store in another table for workforce management by the OME. Faults that are cleared are manually reported by the crew and it is checked as cleared so that the GIS screen will show the network OK at that location. In the map window, locations are shown in green color to indicate that they are functioning properly while locations that have been reported faulty are shown in red (figure 5.10)

The map interface was developed using the GIS ActiveX from MapWindows [4]

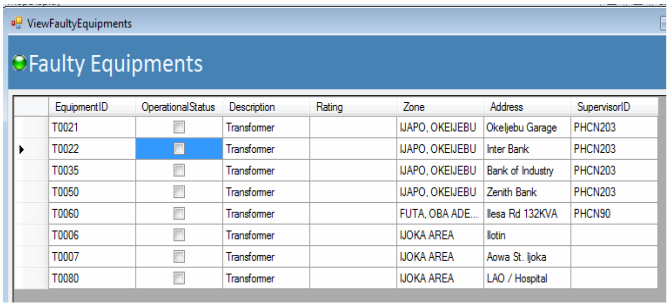

Figure 3. Faulty equipment reported

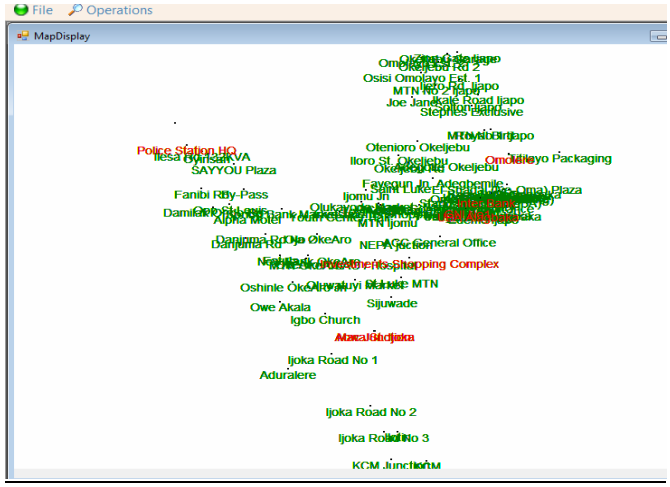

Figure 4. Map display for reports of figure 3.

CONCLUSION

An outage report system was developed using Visual C#. The report was made to the computer using short message service (SMS) of the GSM. This greatly reduce the time and cost of faults on the distribution authority and ensure accurate fault data collection and management.

REFERENCES

1. Chow J. and Rushby J., (1994). Model-Based Reconfiguration: Diagnosis and Recovery", Tech. Rep. 4596, NASA Contractor Report.

2. ETSI, (1992). ETS 300 536 (GSM 03.40), "European Digital Cellular Telecommunications System (Phase 2); Technical Realization of the Sort Message Service (SMS) -Point-to-Point (PP)." European Telecommunications Standard institute, Paris.

3. ETSI (1993) ETS 300 559 (GSM 04.11), "European Digital Cellular Telecommunications System (Phase 2); Technical Realization of the Sort Message Service (SMS) - Point-to-Point (PP) Support on Mobile Radio Interface." European Telecommunications Standard institute, Paris.

4. Daniel P. Ames, 2007. Mapwingis Reference Manual a Function Guide for the Free Mapwindow GIS Activex Map Component. MapWindow GIS Open Source Team