Volume 48, Number 3, 2007

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

REDUNDANCY FOR THE SECURITY SYSTEMS IN A SMART BUILDING USING INTERNET (WEB) AND GSM (SMS) TECHNOLOGIES

Mugur MOCOFAN Radu VASIU Mircea ABUCEANU
Politehnica University of Timisoara, Electronics and Telecommunications Faculty
Bd. Vasile Parvan, Nr. 2, Timisoara, Tel: +40 256 403319, Fax: + 40 256 403300
mugur@cm.upt.ro radu@cm.upt.ro mircea@multimediadesign.ro

Abstract: Security is a problem at multiple levels and for this reason we create an isolated, parallel network to control the smart buildings using the SMS technologies.

Key words: *smart buildings, SMS server, GSM.*

I. INTRODUCTION

What if buildings could function like living systems, altering their shapes in response to changing weather conditions or the way people use them? That's the vision of a new breed of architects who are working on what they think is the future of architecture - "responsive structures" that observe their internal and external environment and change form to suit any situation. A building that mimics a living system would be able to sense and respond appropriately to exterior conditions like varying winds, temperature swings or changing sunlight. Inside, the building might change to accommodate crowd flow or better circulate warm air.

While advances in technology have made interiors more intelligent and comfortable (intelligent lighting, smart air conditioning systems, etc.), the idea of a building as a whole changing shape is only just emerging because of the complexity of the task.

As building automation systems (BAS) that control heat, air conditioning, lighting and other building systems get smarter, they're converging with traditional IT infrastructures. Emerging standards are enabling data sharing between building systems as well as with other business applications, improving efficiency and real-time control over building operating costs. Information security concerns, immature standards, the reluctance of vendors to give up proprietary technologies and ignorance among IT professionals of the convergence trend are all slowing the pace of this transformation, but its gathering momentum.

Facilities managers are driving the change by demanding more-open systems. They are pushing BAS vendors to transform today's closed technologies into Web-enabled applications running over industry-standard IP networks. In addition, the management of BAS is likely to increasingly fall to IT.

"IT folks are entering an era where virtually everything is converging in their direction, and it broadens their horizons tremendously," says Rick LeBlanc, president of HVAC products at Siemens Building Technologies in Buffalo Grove, Ill. IT won't operate BASs, but it will serve the facilities staff as a customer in much the same way it does accounting and other departments today, he says.

Many large companies already have centralized BASs that monitor and control the environment throughout large buildings and across campuses. These systems have begun to migrate to more open IT infrastructures in much the same way that telephone systems and IT networks have converged.

Today's BASs typically include a network of sensors and other devices connected to controllers on each floor, a master controller for a building or campus, a Web server front end for monitoring building systems, and a back-end database for storing historical data. However, as intelligence continues to move into actuators, chillers, security cameras, sensors and other elements of building systems, these devices will increasingly communicate as peers via Web services, allowing BASs to be more flexible and integrate better with other systems.

Open standards are just beginning to evolve and will likely break down the silos between building systems ranging from physical security to elevator controls. In addition, the data from those systems is likely to be shared with other business applications such as the accounting system. This will allow for more-efficient buildings as applications are developed that can capitalize on newly converged data streams and real-time access to data [1].

In the past, controlling the heat involved a call to the facilities person in the basement, who would turn valves to adjust the temperature. Current automated systems use sensors to detect comfort level and actuators to control the valves, but little else has changed.

Standardization has started from the bottom up. Proprietary cabling systems in networks that link sensors and other devices to controllers on individual floors have given way in recent years to two competing, open protocols, BACnet and LonTalk, while floor controllers are migrating onto IP backbones.

The pieces for successful IT/BAS integration are not all in place yet. Various XML groups are developing schemas to interface the building systems to the business systems, but right now, the lack of an industry wide language to program controls is an impediment. XML-based schemas must to evolve but the basic interfaces must come first. The need is an abstraction layer so that programmers or other users don't have to understand control systems.

However, sharing the IP backbone raises security concerns among network administrators. Security is a problem at multiple levels and for this reason, we create an isolated, parallel network for control using the SMS technologies.

II. THE REDUNDANCIES IN SMART BUILDINGS

From the security reasons is necessary to have the redundancy for the signaling and alarm system. The standard internet connection can be broken and the entire control system to be down. In this paper, we propose de redundancy using a SMS system. By SMS is possible to command different actions from the smart home. The system sends alarm, state information and receives from the users commands [2].

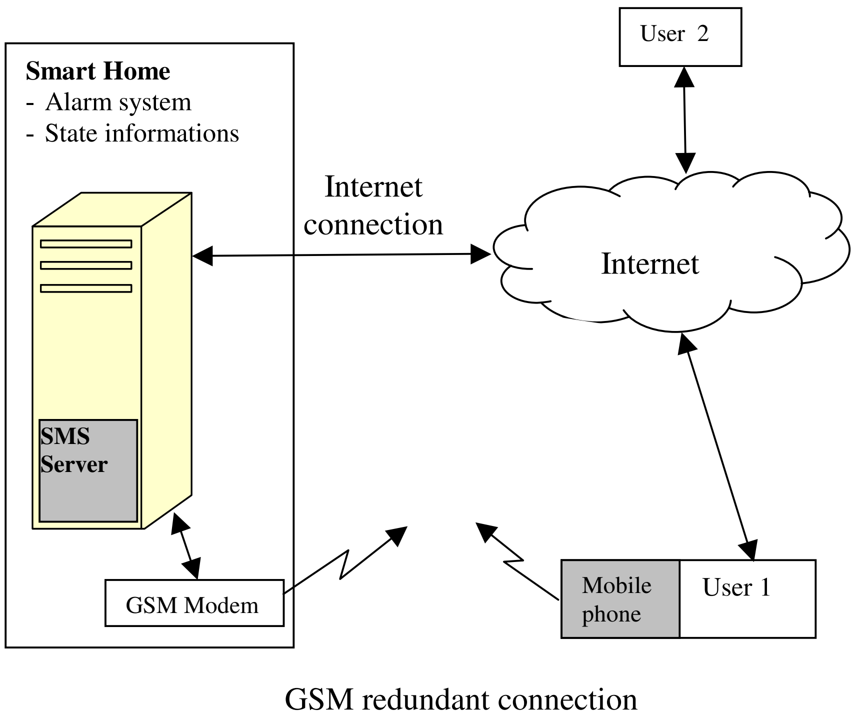

Figure 1. Redundant Smart Home System.

III. THE SMS SERVER

The figure 2 is an overview about the core parts of SMS Server. SMSD is the main program, which starts a sub process for each modem and controls those processes. This example uses 4 modems and it uses the sorting-by-provider function to save money [4].

Steps:

- Sending SM starts by creating a SMS File.
  - The SMS File is a simple text file that contains the destination phone number and the message text.
  - After writing the text, the file will be store in the Outgoing Queue folder.
  - The Outgoing Queue allows creating many files without waiting for the slow modems.
- The next step is to validate the SMS File.
  - SMS Server checks the Outgoing Folder for new files every few seconds.
  - If the destination number is blacklisted, then the file is moved into the Failed Folder.
  - By default, SMS Server moves all successfully checked messages into another queue folder with the name "checked".
- Provider selection
  - But if is use the provider-sorting feature, then the program sorts messages by provider name into one or more queue directories with the name of that provider.
  - Each modem can be attached to one or many Provider Queue folders.
  - SMS Server sub process controls the modem to send short messages.
- Failed messages
  - If sending fails, SMS Server will try a second time. If it fails again, the SMS file will be moved into the Failed Folder.
  - If sending fails 3x 2 times, then the program thinks that the modem is broken and disables it for a while.
- Receiving messages
  - First the modem receives the incoming message and stores it temporarily in its internal memory. Then SMS Server downloads the message from the modem.
  - Messages are stored in text files with random filenames in the Incoming Folder.

SMS Server provides some more functions:

- The Status Monitor shows what your modems are doing. For example Modem1 and Modem3 are receiving messages while Modem2 is sending and Modem4 is idle.
- SMS Server collects statistic data that tell you how

many SMS were sent, received and rejected. Additionally the statistic files give information about the usage of each modem. When you quit or kill SMS Server, it does not loose statistics.

- The Log file tells what is going on. It is specially useful for troubleshooting. SMS Server can also use the syslog or eventlog feature of the operating system.
- Eventhandlers are external programs that can be used to enhance SMS Server. Can be run external programs to validate messages or to perform additional tasks during sending and receiving.
- Alarmhandler. In case of problems, the program generates alarm messages in the logfile and it can call an optional script or program.

Character set used in the language file should match to the locale setting used by the system, which creates outgoing message files. This is very important when UTF-8 is used as a locale, because byte sequences of special characters should match.

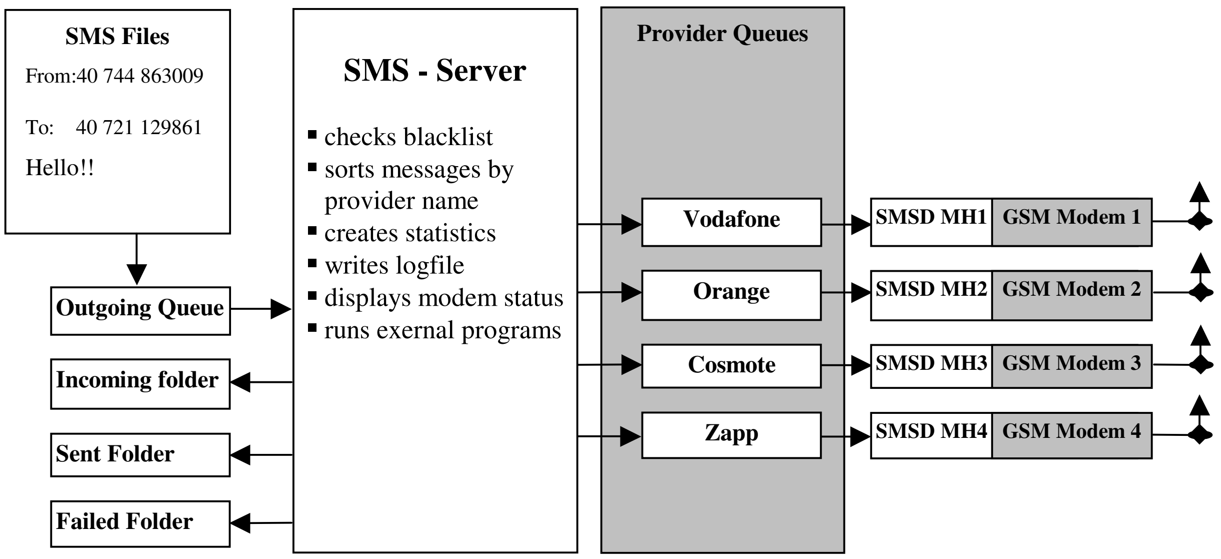

Figure 2. SMS Server

The SMS Server is a SMS Gateway software which can send and receive short messages through GSM modems and mobile phones [4].

The program runs as a SMS daemon, which can be started automatically when the operating system starts. High availability can be ensured by using multiple GSM devices (currently up to 64, this limit is easily changeable).

The program can run other external programs or scripts after events like reception of a new message, successful sending and also when the program detects a problem. These programs can inspect the related text files and perform automatic actions, for example storing information into a database (for example MySQL or Microsoft SQL Server), sending an automatic reply, forwarding messages via email (SMS to eMail gateway) etc.

The SMS Server moves files between spooler directories using the original file name. If outgoing files are created using a fixed filename (which is not recommended) and lot of files using the same name are created within a short period, it's possible that previous file and it's .LOCK file are still existing in the spooler directory. In this case a new file cannot be moved to the spool. Previously the sms server stopped with a fatal error in this case. Now an alarm handler is called and after it has returned a file moving is retried. If a file still cannot be moved, the sms server will stop with a fatal error.

The alarm handler will be use to help with a file moving conflict. The script can wait until a spooler is ready to use, or it can wait some fixed time like 5 seconds. It can also produce some notices to the administration, if necessary.

When a status report is received, it's stored to the incoming folder as described in the SMS file format. An event handler can find the relevant sent message and add the Received: timestamp header to the message file.

If a destination phone is switched off, or it's out of GSM network, the status report is not received, of course because the message is not delivered. We have to check if there is a sent message, which had status report, requested (Message_id field is present) and there is an alternate destination number, but not a Received timestamp. If this kind of message is found, we have to check how long we have been waiting a delivery. For example after about 30 minutes of waiting, we could do the following: Alternate_to number is taken to the memory and removed from the first message.
The modified first message is copied to the temp. To number in the temp message is replaced with a number which was an alternate to. A temp message is moved to the outgoing folder. It's not necessary to remove old Message_id and Sent headers. The SMS Server will remove them automatically when the message is moved to the sent folder. If a status report is requested for the new message, the sms server will place a new Message_id header to the message file.

Configuring Providers: using [queues] and [providers] is optional. If you do not need it, leave both parts out. If you use it, you need to write both parts together into the config file and both parts need to have exactly the same number of lines with the same names. Configuring providers enable a sort function in SMS Server. It takes a look at the destination phone numbers and sorts them into many queue directories - one queue for each phone network provider.

The individual queues allow to assign modems specially to individual phone network providers, which can save a lot of money in some countries.

The name is only a short name for the queue directory. You would typically place the name of the phone network provider here. Spaces and control characters are not allowed here.

The number prefixes are the first digits of phone numbers that belong to the provider. This can be a single number or a comma separated list of many numbers. Write them in international format but without the first "+" character.

IV. SMS FILE FORMAT

An SMS file is a text file that contains the message and a header. Is necessary to store all SMS you want to send in these files in the outgoing directory. The filename does not matter but it has to be unique. Easy example:

```
To: 40 744 863009

Hello, this is the sms.
```

Write the phone number in international format without the leading +. When you like to send a message to a short number (for example to order a ring tone), then precede it with an "s". More complex example:

```
From: MMD
To: 40 744 863009
Flash: yes
Alphabet: ISO

Hello, how are you?
```

Can be added as many header lines is needed. When the program finds an unknown header, it simply ignores that line.

Flash: boolean value. If yes, then the message appears directly on the phones display. Most phones support this feature, but not all.

Alphabet: Tells the program what character set is used in this sms file. Possible values are

- ISO, Latin, Ansi: Normal 8 bit character set, also called Ansi or Latin-1. All three keywords do the same. If the text is longer than 160 characters, then the program can split it automatically. This is the default.
- GSM: 7 bit character set, as described in the GSM specification, with one exception: The @ character is represented by the character code 0xB7. If the text is longer than 160 characters, the program can split it automatically.
- UCS, Chinese, Unicode: UCS2 character set, maximum 70 characters. All three values do the same. The header must be written with an 8-bit character set but the message text part must be written with the 16-bit Unicode (big endian) character set.

The program checks only the first 3 characters of that line, therefore keywords like ISO-8859-15 or UCS-2 will also work fine.

UDH / Only binary messages: Boolean value, tells if the message data contains a user data header. Default is true.

UDH DATA / Only text messages: User data header in hex-dump format. See udh.html and GSM 03.38.

Queue Name of the provider, can be used to override the normal sorting algorithm configured by [providers] and [queues] in the config file.

Report: boolean value, controls if a status report is requested for this message. Without this line, the setting from config file is used.

Autosplit: Controls if and how the program splits large text messages. Without this line, the setting from config file is used. The program does not split binary messages, Unicode messages and text messages with UDH.

- 0 disabled
- 1 enabled, no part-number
- 2 enabled, text numbers
- 3 enabled, concatenated format (not supported by some phones)

Priority: possible value is: high. Message is handled first when moving to spooler and when taking from spooler to sending

Validity: Defines a message validity period. Without this line, the setting from config file is used.

You can specify value as a number following one of the keywords: min, hour, day, week, month or year. Validity period will be rounded down to the nearest possible value.

Voicecall: boolean value, with this feature the sms server will make a voice call to the receivers phone number given in To: header. If the receiver answers to the call, some DTMF tones are played. The message text must start with TONE: keyword. After this there can be number and space, which is number of times to repeat the tone sending. Supported tones are #,*,0...9 and the tone list must be comma separated.

Time: number definition can be used. After a time has reached, hang up is done. If a call is answered before a time is reached, normal sound playing is done. NOTE that this time counting starts after a command is given to the modem and there will be some delay before receiving device starts ringing. You should test this with your own handset to find a reasonable time, which works fine, in the network you are using.

More specifications and details at [3].

The received SMS are stored in the same format as described above but they have some additional header lines. For example:

```
From: 40 744 863009
From_SMSC: 40 744  222333
Sent: 07-02-21 22:26:23
Received: 07-02-21 22:26:29
Subject: modem1
Alphabet: ISO
UDH: false

Text sent with my mobile phone.
```

The filenames of received messages look like modem1.xyzxyz. They begin with the name of the modem that received the message, followed by a dot, followed by six random characters.

The received status reports, if the SMSC and your modem support this feature. Example:

```
From: 40 744 863009
From_SMSC: 40 744 222 333
Sent: 07-09-24 22:26:23
Received: 07-09-24 22:26:29
Subject: modem1
Alphabet: ISO
UDH: false
```

```
Message_id: 234
Discharge_timestamp:       07-09-24
22:27:01
Status: 0,Ok,short message received
by the SME
```

SMS STATUS REPORT

Message_id This is the ID number of the previously sent message, where this status report belongs to. The SMSC gives each sent message such a number.

Discharge_timestamp: this is the time, when the message was successfully delivered or when it was discarded by the SMSC.

Status: The status of the message.

User Data header was added to the SMS format specification to add new features. Originally SMS was made to send single small binary files or text messages, with a maximum of 140 bytes or 160 7-bit characters. But now mobile devices need to distinguish between different files, for example ringtones, operator logos and wap-push messages. Also users want to send larger messages which was impossible with the original SMS format specification.

Each short message has a flag to indicate if the message part includes a User Data Header or not. If this flag is set to 1 (or true), then the first few bytes of the message are the User Data Header, followed by the message text or data. Now you have a more beautiful text-part and the User Data Header is dumped into human readable format, which makes reading it much easier.

The most popular use of User Data Headers are concatenated text messages. If somebody sends a text message that is longer than 160 characters, most phones splits the text automatically into two or more short messages. Each message contains a header that is use by the receiving mobile phone to combine them in correct order.

When concatenated text messages are receive, the parts might arrive in random order. The GSM specification does not force any device to transfer messages in the original order. So when is received a concatenated message is needed to check if all parts have been received before you can put all the parts together.

In case of binary messages, the header is part of the binary data and does not appear in the header, no any UDH-DATA header in binary message files.

IV. EXPERIMENTAL RESULTS

The implemented system contains a general Linux server for all the services (MySQL, APACHE, PHP, FireWall). All the information regarding the sequences of the smart building states are record in a database. This server does the management of all the alarms. In plus the installed SMS Server, is connected with the GSM modem (mobile phone) by a USB 2.0 port. The transmitted/received sms messages are process and the final command are send by a parallel port. With small sms it was possible to control the level of lighting and the temperature. Also, the alarm signals from the security system are sending using the same system.

The next step will be development of the new interface between the smart equipments and database.

III. CONCLUSIONS

From the security reasons is necessary to have the redundancy for the signaling and alarm system. The standard internet connection can be broken and the entire control system to be down. In this paper, we propose de redundancy using a SMS system. By SMS is possible to command different actions from the smart home. The system sends alarm, state information and receives from the users commands.

As extension, the system can be use also for a GPRS connection to the Internet. The better solution is to have a back-up for all type of communication in a smart building. The SMS solution for sending/receiving command can be use in the situation of isolated location, where the GSM cover exists.

REFERENCES

[1] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer – “EasyLiving: Technologies for Intelligent Environments”, *Proc. of the Intl. Conf. on Handheld and Ubiquitous Computing*, pp. 12-27, 2000.

[2] M. Mocofan – “Prevenirea şi evaluarea catastrofelor naturale prin metode moderne de prelucrare a imaginilor” *Raport contract de cercetare CNCSIS*, Contract nr. 33501/17.07.02, Tema 7, cod CNCSIS 17

[3] http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

[4] http://smstools3.kekekasvi.com