Survey on Monitoring Device for Elderly People (i.e Badge)

Priyanka Bhujbal[1] Raksha Borele[2] Prachi Kamble[3] Sonali Kardile[4] Prof. Shrikant Kokate[5]
[1,2,3,4]Student [5]Assistant Professor
[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4,5]Savitribai Phule Pune University, Pimpri Chinchwad College of Engineering, Pune

Abstract— In recent years many elderly patients having lifelong diseases and the patients live alone in the home. Without the help of any person they may not be able to call for help by themselves that they are in critical condition like falling down or suddenly heart attack which may cause effect on his health. So to overcome the above problems a wearable Badge is designed in such a way that the consumption of valuable time which should be given to the patients for treatment is reduces. The system consists of the wearable Badge, Service Centre and the Android Phone. The system consists of Badge detecting events such as 1.Falling down event (FE) 2.Unexpected movements and the motionless events (UE) 3. Going out and coming back home event (HE). It also includes GSM and GPS module which is used for sending the alert messages and tracking the position of the patient is given to the patient family and the doctor by the Android Application.

Key words: Elderly healthcare, Danger detection, fall detection, unexpected behaviour, GPS, GSM

I. Introduction

During the last decades, there has been drastic increase in the population which has now become a social issue. One of the major problems is that the elderly people lives alone in their house, due to which they do not get enough medical treatment on time. On the other hand, due to the unhealthy lifestyle and the huge life pressure in modern society large amount of patients are suffering from chronic diseases, such as hypertension, cancer and health disease, and most of the people are old. We call them empty-nester, elderly patients suffering from such diseases and postoperative rehabilitation patients as special population, especially when they are out of hospital living alone in their home. Once those people encounter this health threatening conditions, like suddenly heart attack or fall down, they feel helpless in such critical conditions and they may not be able to call anyone for help. And it may leads to a grave consequence, even dead or seriously injured. Therefore, automatic dangerous status detecting and alerting system will be very helpful for these people in such conditions. However, most of the current danger detection systems can't be promoted in practice. So for such kind of conditions some latest technology should be developed to overcome. Many of research prototypes either need deploy expensive instruments in a large scale, or need a wearable device for users in multiple motion sensors or biosensors, which result in great inconvenience. Our device targets at cheap and usability, which will be more popular, especially in developing countries.

In this paper, we designed and implement a low-cost, single chip and multifunctional automatic danger detection tool for such people. It comprised of a specially designed sensor chip (named as Badge) and a Service Centre. The Badge is an infrastructure-free by equipped with a GSM and GPS module and its sensing system only integrates a cheap accelerometer. The Badge runs a set of algorithms to detect the following three dangerous events:

1) Fall event (FE);
2) Unexpected motionlessness and unexpected movements event (UE)

When it detects a dangerous event, it will send a short alert message to his family or doctor by its GSM module and track the position of the patient by its GPS module which does not need extra infrastructures. The user or his family or doctor can configure this tool by the website provided by the Service Centre. For example, they can add the various alert contacts for the fall event. And when the Badge detects a fall, it will send alert messages to these people with Android phones. If possible we can add the camera also so as to capture the current position of the patient.

The rest of this paper is arranged as follows: Section II gives the description of our tool's architecture. Fall detection algorithm will be depicted in Section III. Section IV presents the model for detecting the various unexpected movement and unexpected motionlessness. We will be concluding this paper in Section V.

II. System design

A. System Architecture and Application Scenarios

Fig. 1 shows the architecture of the device and the whole system is centered on the Service Centre. At any time, the user or his family can visit the website of Service Centre to change the configuration settings. Configuration can be done on the basic information of the patients, the alert contacts information and the events setting information.

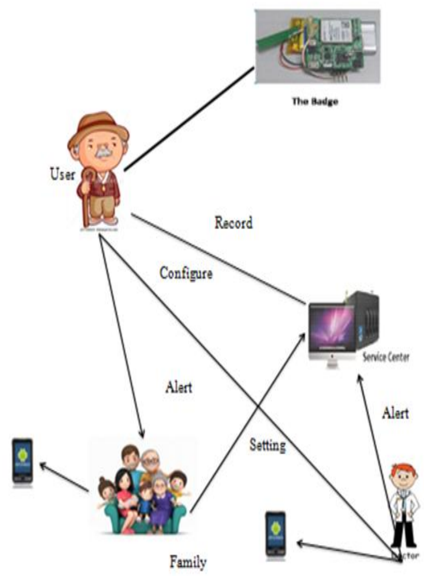


Fig. 1: System Architecture

This entire configuration can be done before the usage of the Badge. The Service Centre will communicate with the Badge through the number provided with the SIM card. With this ID the Service Centre will detect uniquely. The most complex task is the event setting information. The behaviour of the Badge will be changed accordingly. The Badge can be wear by the patient all the time. If the Badge detects the dangerous events then the short alert message will be sent to the patient family and the doctor. And at the same time it will be recorded in the Service Centre so that the data can be use as a reference. At the same time the GPS can be in use for capturing the current position of the patient. So, that the location can used as a reference for the family as well as the doctor.

B. Hardware Architecture of the Badge

Fig. 2 shows the block diagram for the Badge's hardware design. In this design, we use STM8 microcontroller as the processing unit. STM8 is an 8-bit ultra-low power MCU, which is also very cheap. Its core's maximum frequency is about 16MHz and consists of 40 external interrupt sources. It has 64KB of Flash memory and a 4 KB of RAM. Some peripheral were also added for programming and debugging, like SWIM interface, serial communication interface and display device, like LEDs and Buzzer. Flash is a kind of cheap and it is widely used as a storage device. So a Flash, size of 8M, was added for storing the Badge's configuration information. Because we can only add the necessary parameters of the configuration on the Badge, like alert telephone number, event model parameters and some optimized thresholds. For GSM module M35 was selected because of its tiny size and the short message is more flexible and power saving than any other communication technologies. It can be easily controlled by the STM8 for sending and receiving messages.

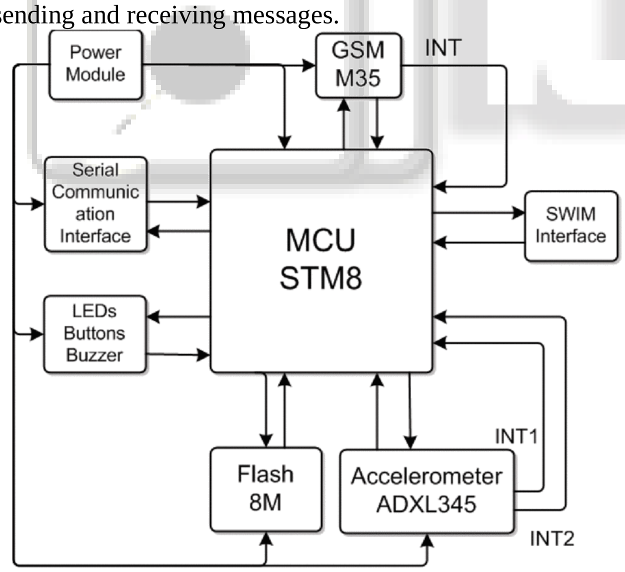


Fig. 2: Hardware Design of the Block Diagram

Accelerometer is vital components of the Badge. We select the ADXL345 as our accelerometer; because of its highest sampling rate is 3200Hz. But as per the rate of conversation of energy, the higher rate means the more energy consuming and we think 20Hz is more enough. The task of Badge is detecting FE, UE and HE, according to the collected acceleration data. The ADXL345 consist of two interrupt interfaces and eight types of build-in interrupts, such as Data Ready, Free Fall, and Activity and so on. We will use both the inner interrupts and acceleration data to achieve our goal.

C. Software Architecture of the Badge

The software architecture is shown in Fig. 3. It contains three components: the Badge's software architecture, Android Phone and the Service Center's software architecture.

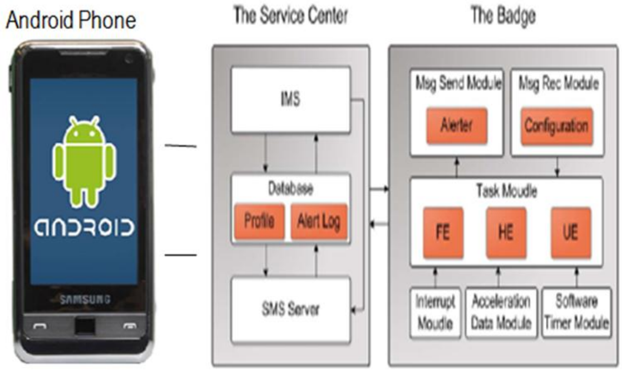


Fig. 3: Software Architecture of the Badge

The vital components of the Badge are the three task modules: FE module, UE module and HE module. FE module encapsulates our fall detection algorithm. It is driven by the inner interrupts from ADXL345. Meanwhile, FE module will read the current acceleration from the Acceleration Data module to get the posture of the user. The Acceleration Data module collects acceleration data constantly and we select 10 seconds as time window to cache those data. In addition, it also has a filter to eliminate data consisting noise. UE module's main task is to detect unexpected movement and unexpected motionlessness. The data cached in time window of the Acceleration Data module will in used to obtain the information it needed. Software Timer module is also an important component. The STM8 only has four inner hardware timers, which are not enough for our tasks. So we create more software timers based on one of the hardware timers. All of the three task modules are controlled by Configuration. When receives a message form Service Center, the Configuration module will parse this message and change the relevant parameters in it. Certainly, the corresponding task's behavior will be changed simultaneously. At last, any event detected will be sent to the Alerter. Firstly, the Alerter loads the relevant alert contacts about this event from the Configuration. Then it sends alert messages to them and reports to the Service Center meanwhile.

The Service Center is constructed on the J2EE platform. Simply, it consists of three parts, as shown in the Fig. 3. IMS (Information Management System) is website designed for users for configuring their devices. SMS (Short Messaging Service) Server is a message processing server, which is comprised of a server and several GSM modems. SMSLib is one of the open source SMS messaging library and provides a universal texting API. It has included in our SMS Server and acts as the messaging driver. When configurations are submitted on IMS, they will be saved into database and some necessary parameters will be sent to the Badge by the SMS Server. While receiving a message from the Badge, the SMS Server parses this message and log this events into database. The user's family or doctor also can browse the user's alert history on IMS.

III. FALL DETECTION

Mostly algorithms are based on FREE FALL + IMPACT DETECTION + ORIENTATION MONITORING, which is shown in Fig. 4. These algorithms first detect the FREE FALL by observing the acceleration lower than a threshold value, followed by detecting an impact. At last, they detect if the 3D-accelerometer's orientation changed is larger than the threshold value. Some algorithms also constrain the time interval between two events performed. For example, IMPACT is valid only when happened following FREE FALL within the time frame of 1 s.

The accelerometer ADXL345 consists of eight useful interrupts. Out of which three interrupts can be used in our algorithm: FF, ACT and INACT. FF is Free Fall interrupt, is triggered when any axis of the 3D- acceleration continues lower than a threshold (FF_THRESH) for the time (FF_TIME). ACT interrupt is triggered when any axis of the 3D-acceleration exceeds the threshold (ACT_THRESH). When all the three axes of the 3D- acceleration changes continue lower than the threshold (INACT_THRESH) over a time threshold (INACT_TIME), it will produce an INAC interrupt. All of these thresholds can be changed in the program. In our tool, these thresholds will be calculated according to the user's physical characteristic on the Service Center and the user can change these parameters on the Service Center by hand if he encounters too much false alerts accordingly. Our fall detection algorithm is based on processing the interrupts sequence produced by the ADXL345 instead of processing the raw acceleration data.

For eliminating more errors, we also consider the posture detection and time interval constraint into consideration. The input set of our algorithm is comprised of this interrupt set {FF, ACT, INACT}, the posture set as {Standing, Lying} and three timers. According to the acceleration readings of a fall presented in Fig. 5, we create four states to represent each process respectively: NORMAL, START FALL, FALL GROUND and FALL. One highlight of our algorithm is that when detect a FALL, the user's posture will be continue monitored. If the user is lying on the ground for a long time, we will think he may be faint or seriously injured caused by the fall event. So the MOTIONLESS state is added into our algorithm. Another highlight of our algorithm is that we also allow for the series of falls events.

We also use a timed automation to describe our algorithm. The initial state is the NORMAL. Transition A means when the FF interrupt was triggered, a fall may be happening and we come to know the START FALL. At the same time, three timers will be restarted, Timer1 (t1) will wait for the next FF interrupt, Timer2 (t2) will wait for an ACT interrupt and Timer3 (t3) will wait for an INACT interrupt. Transition G is taken into consideration of the continuous falls. When the series of falls happens, it may produce several continuous FF interrupts. So if an FF interrupts is triggered on the START FALL before the timer1 is over. Otherwise, the FF_COUNTER will reset to 1. And when the INACT is triggered before the Timer3 is over and if the FF_COUNTER is larger than the threshold and user's posture is lying then the event SERIES FALL is detected. Transition H means going back to the NORMAL state. Transition B is activated only when the ACT is triggered before the Timer2 is over. It may indicate that user fall down and may hit the ground, and then we come to the FALL GROUND state and restart the timer for the INACT interrupt. Transition I is similar to the Transition H. Transition F is for avoiding multiple impacts within the period of falling down. And if the ACT or FF is triggered on the FALL GROUND state and user's posture is lying then we will restart the Timer2 and stay on the FALL GROUND. Also when INACT is triggered and the Timer2 in not over then fall is detected and we came to now that it is fall event. But this is not the conclusion we will still continue to monitor the users posture. The user is sustained with inquiries of loss of consciousness if the user lies on the ground over the threshold. And if the user gets up after the fall then the transition E will be activated, then we report the fall happened is normal. The following are the figures which tells us about the actions events happened or going to happen with the help of the above algorithm.

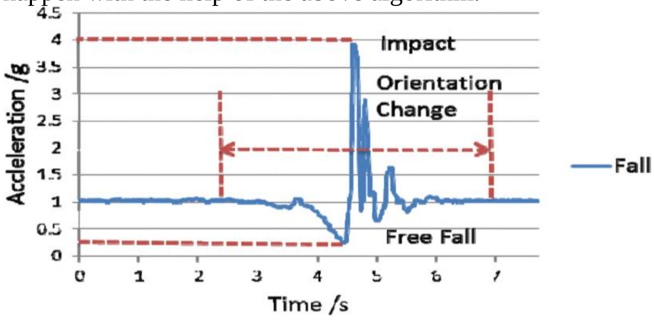
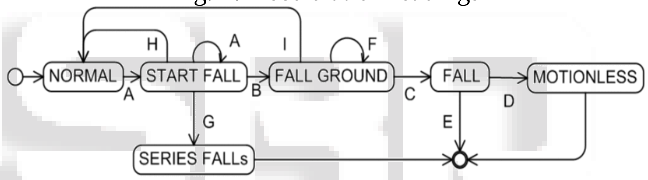


Fig. 4: Acceleration readings



A) FF[t1:=0,t2:=0,t3:=0]; B)ACT & t2<T2[t2:=0]; C)INACT & posture=Lying & t2<T3 [t2:=0]; D)posture=Lying &t2>T4; E)posture=Standing; F)(ACT | FF) & posture=Lying & t2<T3 [t2:=0]; G)INACT & FF_COUNTER>MIN & posture=Lying &t3<T4; H)t3>=T3 | (INACT &(FF_COUNTER<=MIN | posture=Standing)); I)t2>=T3 | posture=Standing

Fig. 5: Time automation for fall detection

IV. UNEXPECTED AND UNEXPECTED MOTIONLESS DETECTION

Abnormal behavior often indicates that something is wrong with the health. Equipped with the Badge, our tool can do this abnormal behavior detection: unexpected movement and unexpected motionlessness, such as too much movement when asleep (Bad Sleep) or long motionless when it's time to get up and no response (No Get Up).

A. *Example*

Now, here we introduce our model with an example. Generally, we try to create an instance by our model to detect No Get Up event. First of all, we assume that the user gets up before 9 o'clock normally. If he did not get up before 9 0' clock, we may think that something is wrong and No Get Up is detected. Then we can set the Effective Time range from 9:00 to 10:00. Effective Time means only in this period we will do the detection works. The Posture here means the angle between the human body and the vertical direction. When the Posture is larger than the 70 degree, it means the user is lying flat. On Service Center, we have divided the magnitude of Activity Frequency into three levels: VAR_SMALL, VAR_NORMAL and

VAR_LARGE. There is no doubt that the Activity Frequency should always be less than VAR_SMALL when sleeping. The Duration can set to 10 minutes. Then we can add this model instance into the Badge by Service Center. Every day, starts from 9 o'clock, this instance is been activated. If it finds that the user is still lying, motionless and continues more than 10 minutes, a No Get Up is detected. Then the Badge will send the message to the contact is stored.

Attribute	Definition		
	Computing method	*Upper bound*	*Lower bound*
Acitivity Frequency	$\sum \lvert A_{i+1} - A_i \rvert$	VAR_UP	VAR_LOW
Posture	$\cos^{-1} \frac{A_s \cdot A_n}{\lvert A_s \rvert \lvert A_n \rvert}$	ORI_UP	ORI_LOW
Duration	Inner timer.		DUR
Effective Time	Inner RTC (Real-Time Clock)	T_BEGIN	T_END

A_s is the acceleration vector when standing.

Table 1: Model Definition

B. Method

As mentioned before, we designed a model to define what an unexpected behavior is. The model mainly contains four Attributes: the Activity Frequency, the Posture, the Duration and the Effective Time, as shown in the TABLE I. The Activity Frequency here is obtained from the 10 seconds time window by accumulating the difference between the two contiguous acceleration vectors. The Posture can be computed from the acceleration vectors of the Normal State as well as the current state. The Duration constrains the minimum time of the unexpected behavior, which is timed by inner timer. The Badge's clock is the inner RTC and it will be calibrated by the Service Center. Therefore, when comes to the Effective Time, if we find that the Activity Frequency is within the range between VAR_UP and VAR_LOW and the current Posture is between ORI UP and ORI_LOW, then this time window is the potential part of the abnormal behavior and we added it to the unexpected behavior analyzer. When the analyzer finds the several contiguous time windows of the unexpected behavior and their sum of time beyond the DUR, then an unexpected behavior defined by this model instance is been detected. Here the contiguous time window has not to be adjacent closely, all it needs is their intervals cannot be larger than DUR. Besides, every instance of the model has a unique ID to distinguish with the others and it also must associate the contact for alerting.

In the Badge, we already store all the unexpected behavior model instances in the Flash. And they are loaded into the linked list when the Badge is running. Certainly, these model instances can also be edited on the Service Center. So, users can change the parameters of them or even add new instances accordingly.

V. CONCLUSION

In this paper we have presented the We Care tool for the automatic detection of the danger. Our tool is consist of the wearable chip (the Badge) and the Service Center. The Badge is simple, cheap and easy to use. With an accelerometer, it can detect the various events like falling down event, the unexpected movement and motionlessness events and the going out and going back home events. A novel unexpected movement and motionlessness detection model is also proposed, which is useful and more flexible. For capturing the going out and going back home events, we need to recognize the elevator taking and stair climbing behaviors. Although the accuracy of recognizing stair climbing is considerable lower when climbing only one floor and we cannot distinguish the climbing directions automatically, it has little impact on our functions. A GSM module is also integrated into the Badge for sending the alert messages and receiving the configuration commands from the Service Center without any extra infrastructures. By accessing the Service Center's website, the users can configure their tools remotely at any time. A GPS module is also integrated so as to find out the current position of the patient without any delay. If possible camera can also be integrated so as to capture the current situation of the patient.

REFERENCES

[1] Nadia Zouba, Francios Bremond, Monique Thonnat: An Activity Monitoring System for Real elderly at home: Validation Study, 2010 seventh IEEE International Conference.
[2] Hongwei Xie, Xianping Tao, Jian Lu: Wecare: An Intelligent Badge for elderly danger detection and Alert, 2013 IEEE 10th International Conference.
[3] Nagendra Kumar Suryadevara, Shubhas Chandra Mukhopadhyay: Wireless Sensor Network based home monitoring system for Wellness Determination of elderly, 2012 IEEE SENSORS JOURNAL, Vol. 12, No. 6, June 2012.
[4] Chien J R C, Tai C C: A new wireless type physiological signal Measuring System using PDA and Bluetooth Technology, 2005
[5] The data sheet of ADXL345: http: //www:analog.com/static/imported file /data_sheets/ADXL345.pdf
[6] SMSLib: a universal API for SMS messaging: http://smslib.org//data_sheets/ADXL345.pdf
[7] FALL detection algorithm: http://www.hindawi.com/journals/jam/ 2014/896030/
[8] Liang Y, Zhou X, Yu Z, Bin G: Energy efficient motion related activity recognition on mobile devices for pervasive healthcare , 2013:1-15

