Volume 10 Issue 04 Published, November 9, 2006 ISSN 1535-864X DOI: 10.1535/itj.1004.06

Intel® Technology Journal

Autonomic Computing

A Self-Managing Framework for Health Monitoring

More information, including current and past issues of Intel Technology Journal, can be found at:
http://developer.intel.com/technology/itj/index.htm

A Self-Managing Framework for Health Monitoring

Amit Baxi, Corporate Technology Group, Intel Corporation
Nagaraju Kodalapura, Corporate Technology Group, Intel Corporation

Index words: Body Area Network, Zigbee, patient monitoring, autonomic, self-management, power management, fail-safe mechanisms, biometric authentication

ABSTRACT

Existing medical devices and equipment monitor abnormal physiological conditions and trigger audio-visual alarms. However, these devices require high levels of interaction by a doctor since they do not have any self-managing capabilities to reduce user intervention or to autonomically handle alerts.

Recent advances in low-power wireless communication technologies have led to the development of small form-factor, wireless, body-wearable biomedical devices for health monitoring [1]. This enables patient mobility and comfort and enables continuity of care from hospital to home. In this paper we propose a self-managing framework for health monitoring using body-wearable bio-devices [2] that can reduce the doctor's intervention for patient management. This will enable doctors to effectively manage a greater number of patients and reduce the number of errors inherent in paper-based processes.

We illustrate three usage models where this self-managing framework can be applied: remotely monitoring patients at home, monitoring fetal well-being in a maternity ward, and monitoring critical patients in an Intensive Care Unit (ICU).

INTRODUCTION

There has been a significant development in bio-medical instrumentation over the last two decades. A number of high-tech medical diagnostic and therapeutic systems are used by doctors for diagnostics, for monitoring critical patients, and for delivering treatment. Such devices detect abnormal physiological states, and they trigger audio-visual alarms when the limits are crossed. However, they require frequent intervention by doctors to manage these alarms.

If such patient monitoring systems could be equipped with some intelligence and self-managing capabilities, the systems would be able to autonomically handle several alarm conditions for efficient operation, would require less intervention by doctors and would reduce errors related to patient management.

In this paper we describe a self-managed generic framework for digital health monitoring that can be applied in different use cases. The self-managing capabilities of such a framework are useful to efficiently manage several tasks that usually require user intervention.

We believe that such a system design will enable medical device manufacturers to design their biomedical front-end devices to be compatible with a generic framework and this will drive standardization of hardware and software interfaces for medical devices.

Moreover, this framework utilizes the advances in Personal Computer (PC) and server technology to provide reliable, generic, cost-effective and efficiently self-managed patient monitoring solutions.

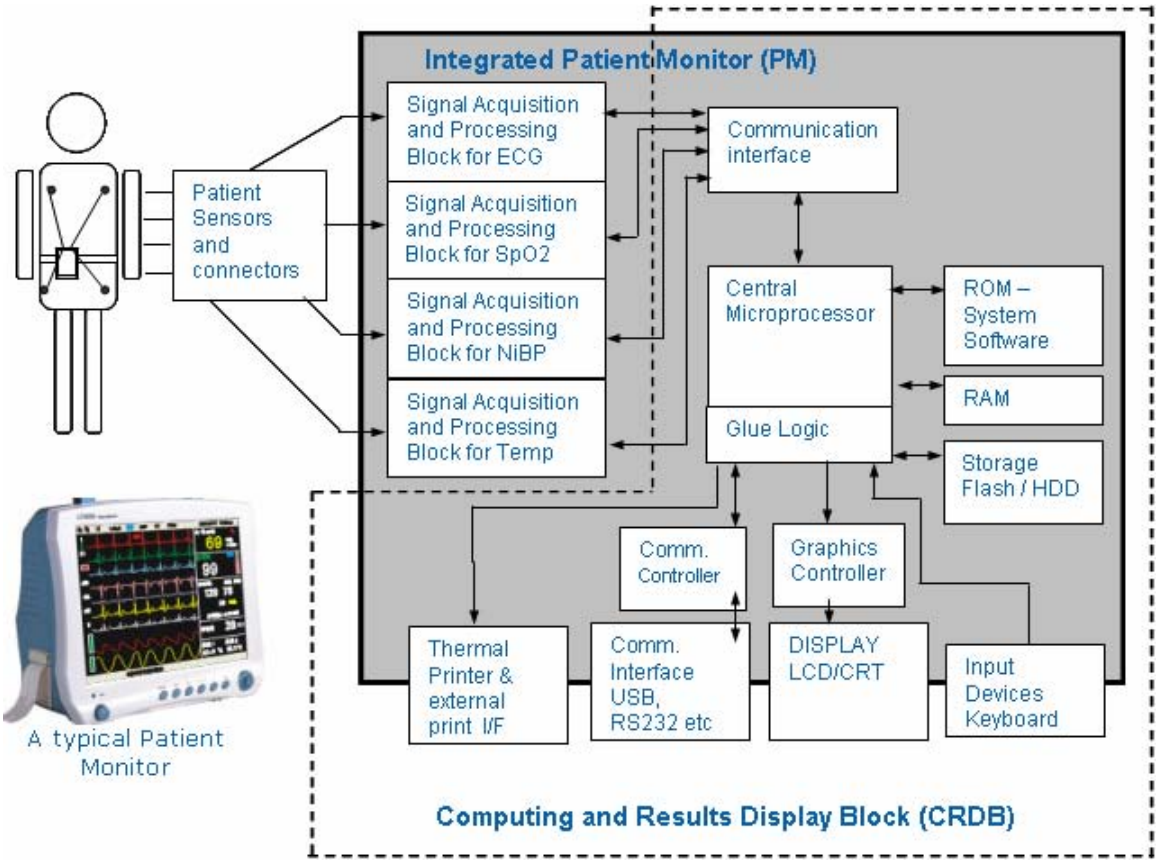

Figure 1: A conventional multi-parameter patient monitor

OVERVIEW OF EXISTING PATIENT MONITORING SOLUTIONS

Patients in the Intensive Care Unit (ICU) are usually wired to a bedside patient monitor known as a Multi-Parameter Patient Monitor (MPM). The MPM monitors one or more of the patient's vital physiological signals such as Electrocardiogram (ECG), Blood Oxygen Saturation (SpO2), Non-Invasive Blood Pressure (NiBP), Invasive Blood Pressures (IBP), Respiration, Temperature, Airway Gases, etc. Patient Monitoring Systems (PMS) are usually dedicated embedded systems, each with its own sensors, biomedical front-end hardware, LCD display, keypad, integrated thermal printer, communication interfaces such as RS232 or USB, and battery backup. These systems are quite expensive and thus contribute to the overall high costs of healthcare. These dedicated embedded systems have proprietary hardware and software. The biomedical Original Equipment Manufacturers (OEMs) avoid the use of general-purpose PC platforms for critical applications since the PC's hardware, software, and operating system (OS) are not optimized and reliable enough for medical applications. A typical MPM is shown in Figure 1.

Patient monitors have biomedical signal processing algorithms for detecting abnormal physiological conditions such as cardiac arrhythmias, apnea, low blood pressure, etc., and the monitor usually gives an audio-visual alert when such a condition is detected. The doctor usually intervenes when an alert is generated and either silences the system or takes corrective action to treat the patient. Apart from alerts generated by physiological conditions, other alerts are also generated to notify system status or malfunctions such as a low battery condition, sensor coming off the patient, etc. Since a doctor managing an ICU has to respond to alerts from several patients, it becomes an uphill task for the doctor to manage alerts, if the patient monitoring system does not have the capability to self-manage such alerts.

Our proposed framework addresses these issues by embedding intelligence at various levels in the system, in order that the system can manage itself and only require intervention by a doctor for critical physiological alerts.

As seen from Figure 1, the Computing and Results Display Block (CRDB) in a typical MPM is quite similar to that of a general-purpose PC. Hence, it makes sense to functionally separate the CRDB from the integrated PMS

and use a single PC to cater to the computing and results display requirements of multiple patients.

A SELF-MANAGING FRAMEWORK FOR HEALTH MONITORING

Architecture

The evolution of low-power wireless communication technologies like Zigbee (IEEE 802.15.4), low-power Bluetooth*, etc. has enabled the development of small, body-wearable, wireless sensors for patient monitoring.

As shown in Figure 2, these sensors can be configured to form a Wireless Body Area Network (WBAN) [3] and enable monitoring of multiple bio-parameters (such as ECG, Pulse Oxygen saturation, Blood Pressure, etc.) of multiple patients at a central location. Each body-wearable sensor, known as a Bio-Front End device (BFE), is composed of a sensor for bio-signal sensing, the related front-end hardware, a low-power microcontroller for data acquisition and a wireless transceiver for data transfer to the receiver. There can be multiple BFE devices connected to a patient for monitoring multiple parameters. An aggregator (AGG) device worn by the patient performs the function of receiving the wirelessly transmitted data from multiple BFE devices connected to a patient and transmits the aggregated data to a backend PC or server. The AGG device can be a device like a Portable Digital Assistant (PDA) or a scaled down version of that without a Liquid Crystal Display (LCD). The AGG and the BFE devices form a localized WBAN, with each BFE device having a unique device ID.

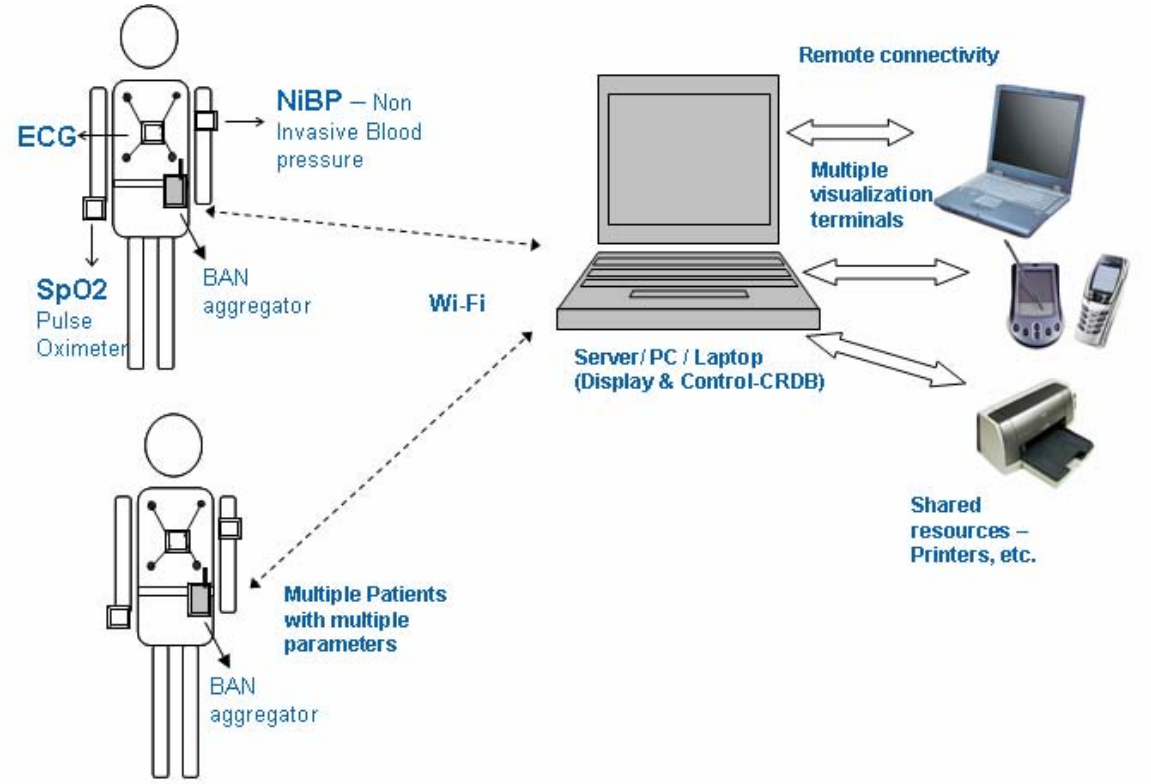

Figure 2: Architecture of a body-wearable, wireless health monitoring platform

Following are the advantages of this architectural framework:

- The functional partitioning of the PMS allows a single PC to cater to the computing and results display requirements of multiple patients, thereby reducing the cost of the overall system.
- There is better resource utilization since a single resource (like a network printer) can be shared across multiple patients.
- It enables patient mobility and comfort.
- It allows central monitoring of multiple patients from a single location.

- It allows the entire patient-monitoring framework (consisting of WBANs of multiple patients) to be managed from a central location.
- Such a framework does not require the doctor to be in close proximity to the patient or to the monitoring equipment, since the patient's physiological data can be made available to the doctor anytime and anywhere on his hand-held communication device.

System Components in the Health Monitoring Framework

There are three major components in the health monitoring framework as shown in Figure 3:

- The BFE device.
- The AGG.
- The backend computational platform such as a PC or server.

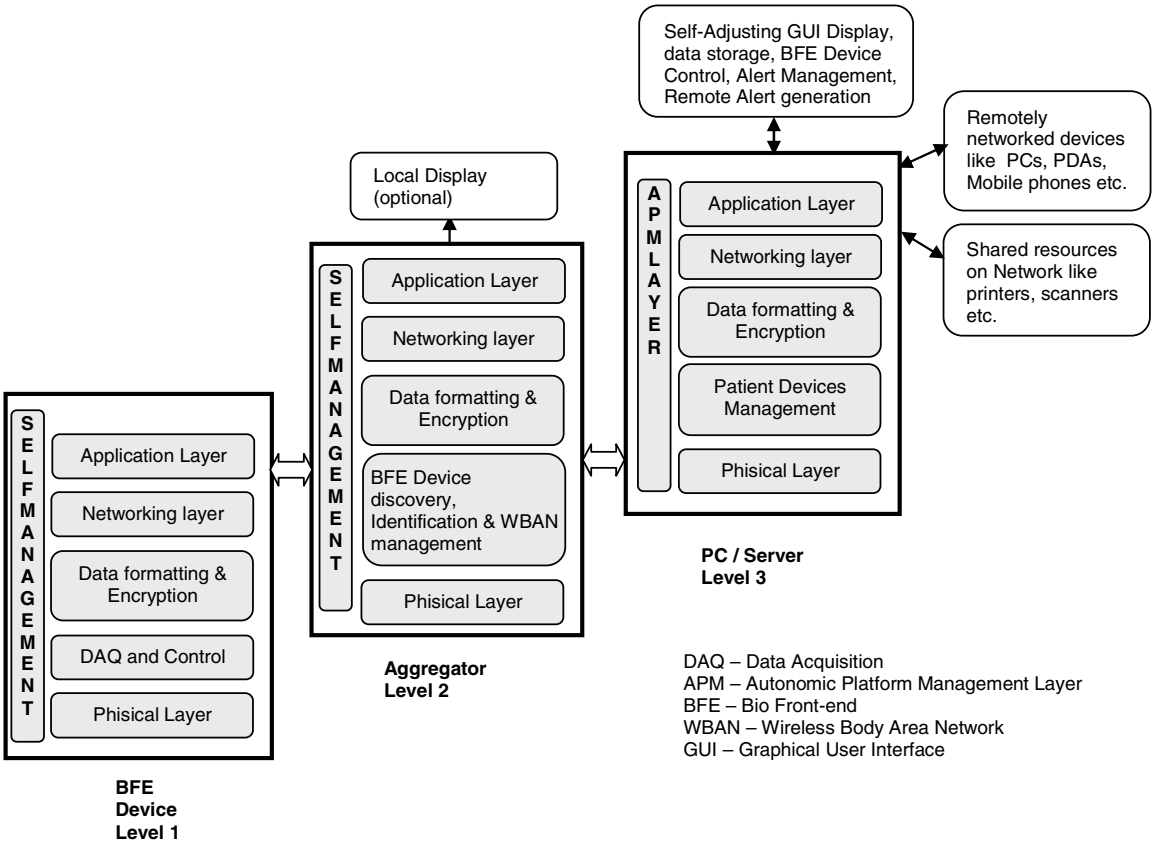

Figure 3: System components of a self-managing framework for health monitoring

The Bio-Front End

The BFE device, which is worn by the patient, consists of the sensor, the front-end hardware and firmware for processing the sensor signals, a low-power microcontroller and a low-power wireless transceiver device for communication. The communication technology between the BFE and the AGG device can be based on a low-power wireless standard like the 802.15.4 (Zigbee) or low-power Bluetooth in order to maximize the battery life. The type of sensor and the front-end hardware is specific to the physiological signal being measured. For example, if the BFE device is designed to monitor ECG, the sensors would be pre-gelled electrodes, whereas for pulse oximetry, the sensor would be an optical finger probe. The microcontroller controls the overall functionality of the device and performs the functions such as data acquisition, device control, data formatting, packet forming, wireless transmission and execution of the commands sent by the AGG device.

The Aggregator

The AGG device is also worn by the patient and it serves to aggregate the data from multiple BFE devices connected to a patient and sends these data wirelessly to a PC for further processing. The AGG uses low-power wireless technology like Zigbee to communicate with the BFE devices and Wi-Fi or other appropriate communication technology to communicate with the PC,

since the required data rate and range is larger than that for a body-area network. The AGG device also serves as the intermediate messaging link between the BFE device and the PC.

The Personal Computer or Server

The PC is used to collect data from multiple AGG devices connected to multiple patients and process these data. The PC also serves to display the physiological waveforms and parameters of several patients on the screen, does further signal processing and computation, performs feature extraction, stores data, and communicates with other terminals over the Internet protocol. The PC also keeps the doctor updated on the patient status by sending the patient information on his mobile phone or PDA. It also manages resources such as printers, scanners, and other devices, which are shared across multiple patients.

SELF-MANAGING THE HEALTH MONITORING FRAMEWORK

The flow of events in self-managing a device in the framework is shown in Figure 4. Self-management involves sensing or monitoring parameters from the device or the system to be managed and then analyzing the parameters to find out whether a corrective action is required. Intelligent algorithms and decision trees are used to decide if a management action is required and then to arrive at the most effective decision to act on the inputs. The decision is then executed by the execution engine in a feedback loop to effectively manage the device or the system without the need for user intervention.

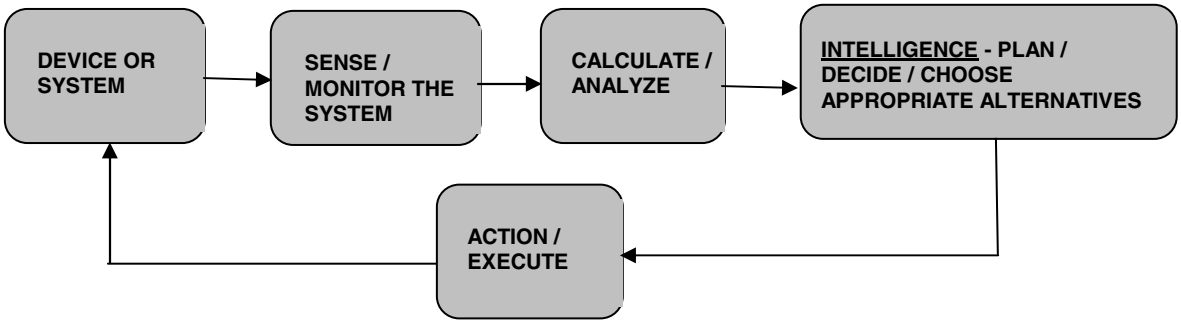

Figure 4: Flow of events in self managing a system

The proposed health-monitoring framework can be efficiently self-managed by building limited self-managing capabilities in each building block and shifting major intelligence and decision-making capabilities to the PC.

The PC makes the control decisions based on status inputs from the connected BFE and AGG devices and then instructs the devices to perform certain self-management functions.

Following are the advantages of this self-management approach:

- BFE devices may not have high computational capability. Hence, they can send their status information to the PC and use the processing power of the PC to make optimal self-management decisions.
- The PC is the central hub of information gathering and it is aware of the status of all devices in the network. Hence, the PC is better equipped to make optimum decisions by considering the status of the entire platform as a whole, rather than the status of a single device.

The BFE and the AGG devices should have limited self-managing capability in order to manage themselves in case the communication link to the PC fails. Such devices can have fail-safe mechanisms and recovery algorithms to put the device in a safe state in the absence of PC connectivity.

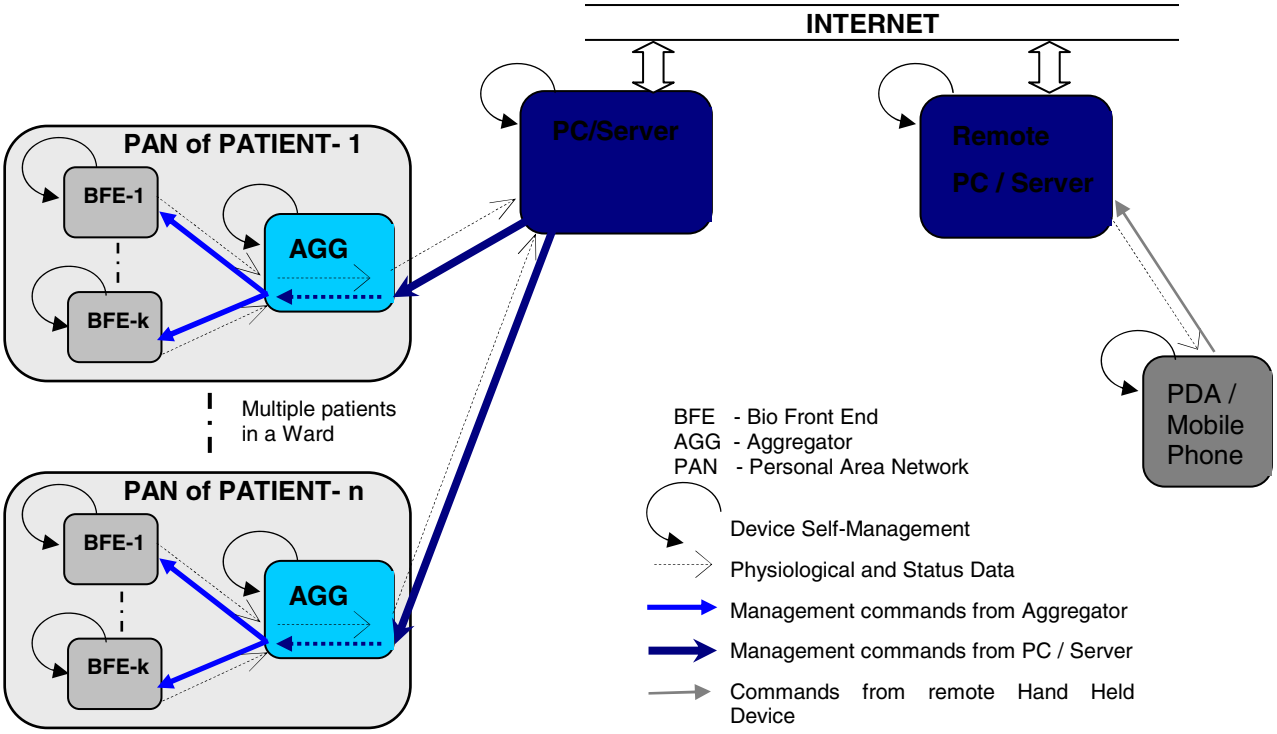

Figure 5: Self managing framework for health monitoring

Figure 5 shows a self-managing framework for health monitoring. Each BFE device connected to the patient self-manages itself to some extent and can also be managed by its WBAN AGG device and by the back-end PC/server. The management scope of a BFE device is limited to itself. The management scope of an AGG device is limited to itself and also to manage all the BFE devices in the WBAN of that particular patient. The PC is the central hub of information gathering and does the system-level management, making decisions based on inputs from all the devices in the system. The management scope of PC extends to itself, all AGG devices in its network and all the associated BFE devices. The PC also interacts with other servers that are remotely connected via the Internet link. The remotely connected hand-held devices and servers can be used to manually manage the BFE and AGG devices connected to the network after secure authentication. The PC manages and communicates with the BFE devices in the network via the intermediate AGG devices.

The self-managing capabilities of different blocks of the framework are described below.

Self-Managing the BFE Device

The self-managing capabilities of the BFE device may vary according to the type of the device. An ECG BFE device may self-manage itself differently than a SpO2 BFE device. Typically, a BFE device can exhibit the following self-management capabilities:

- *Device to patient binding*. This involves binding all the BFE devices and the AGG device connected to a patient to a unique patient-ID and a unique Personal Area Network (PAN) identification (ID) number. The PAN-ID of each WBAN is closely coupled to the Patient-ID (i.e., the patient registration number assigned to the patient by the hospital). The unique Patient-ID and the patient history may be programmed in a small Radio Frequency ID (RFID) wrist strap attached to the patient, when the patient is admitted to the hospital. The BFE devices may have RFID reading capability to read the Patient-ID and be associated with the unique Patient-ID, when they are brought into close contact with the patient's RFID wrist strap. Once each BFE device and AGG device gets associated with the Patient-ID, all the connected devices can form a unique PAN-ID for the patient.

Alternatively, each BFE device may have the capability to autonomically identify a patient based on some biometric technique. Data packets sent by the BFE devices to the AGG device and to the PC are tagged with the unique Patient-ID in order to bind them to the patient's individual database.

- *Network association.* All the BFE devices and the AGG device connected to a patient are programmed to have a unique PAN-ID for their network. As mentioned above, the PAN-ID can be programmed into the BFE devices and AGG device before connecting them to a patient. When the BFE device is turned ON, it looks for beacons sent by the AGG device to check the AGG device's PAN-ID and gets itself associated with the network if it matches with its own PAN-ID. The AGG in turn identifies the newly joined BFE device, determines the type (i.e., ECG, SpO2 etc.) of the BFE device and intimates the same to the PC. The PC dynamically adjusts the Graphical User Interface (GUI) on its screen to effectively optimize the screen space to display the signals from the newly networked BFE device.
- *Self-configuration*. When a BFE device is turned ON it automatically configures itself to a default state. It also checks the attached sensors and re-configures itself for its mode of operation. For example, for an ECG BFE device, there are two modes of operation: diagnostic and monitoring. In diagnostic mode a 10 lead cable is used to sense the ECG signals and the ECG signals are digitized at 500 samples per second per lead. However, in the monitoring mode, a 3 or 5 lead ECG cable is used to sense the ECG signals and the ECG signals are digitized at a sampling rate of 200 samples per second per lead. At start-up, the BFE device can identify the type of ECG cable connected at its input and automatically adjust its mode of operation, the number of channels to be sampled, the sampling rate per channel, and it can reserve the bandwidth required for the wireless data transfer.
- *Self-calibration and self-check*. At power ON or when requested by the PC, the BFE device can also execute self-calibration and self-check routines to dynamically adjust its settings for maximum accuracy. For example, an ECG device may input a fixed reference voltage to its Analog to Digital Converter (ADC), convert it to a digital representation, and then adjust its gain and offset errors to compensate for the errors in the ADC. Similarly, a SpO2 device may calibrate its optical sensor characteristics, whereas, a NiBP BFE device may calibrate its air pressure sensor and self-check its air release solenoid valves and the air pump for proper operation.
- *Power management*. The BFE device can dynamically turn OFF the idle peripherals when not in use. For example, the BFE device microcontroller can turn OFF its communication ports, data acquisition, and control peripherals when not in use in a particular mode of operation. Battery capacity monitoring can be done at periodic intervals by the BFE device and the battery capacity information can be sent to the PC via the AGG device.

  The BFE device can reduce the clock frequency of its microcontroller and related peripherals if too much computing is not required by the BFE device sensor. Very low data rate BFE devices such as body temperature sensors can enter sleep state between periodic measurements, whereas, the medium data rate devices such as SpO2 may turn OFF the wireless transmitter's RF circuitry between data transmissions in order to conserve battery power.

  The BFE device may either execute certain power management functions autonomically or execute them when instructed by the PC. For example, when the battery voltage falls below a certain threshold value, the PC may instruct an ECG BFE device to reduce the ECG sampling rate, and the PC may interpolate the sub-sampled ECG data for proper visual representation. Alternatively, depending on the status of other sensors attached to the patient, the PC may instruct the ECG BFE device to reduce the number of ECG signals to be sampled and reduce the amount of wireless data transmission to conserve battery power.

  Near the end of battery capacity, the PC and the BFE device may issue audio and visual alerts to inform the hospital staff to change the battery. The BFE device also intimates to the PC its decision to turn OFF to a safe state when the battery capacity reaches 0%.
- *Fail-safe mechanisms*. It is very important for the BFE device to have intelligent algorithms to handle alerts in case of malfunctions and put the BFE device in a safe state to avoid patient injury.

  Some important malfunctions could be microcontroller/hardware malfunction, sensor malfunction and communication failure.
- *Hardware malfunction*. These types of malfunctions can be induced by failure of electronic components or can be due to effects of electromagnetic interference (EMI). EMI can cause the BFE device microcontroller to enter a runaway state that can be very dangerous. This can be mitigated by ensuring that, in the case of malfunction, a watchdog timer running inside the microcontroller device either resets

the microcontroller or puts the microcontroller and associated hardware in a known safe state.

In the case of BFE devices such as a NiBP device, the blood pressure is measured by occluding the brachial artery by inflating a cuff tied to the patient's arm. If the air inside the cuff is not released due to hardware malfunction, the blood flow to the patient's hand stops and may cause serious tissue damage. Hence, it is very important to have either a redundant microcontroller or some standby mechanical devices (such as normally open solenoid air release valves) to ensure air release even in case of power or hardware failure.

- *Sensor malfunction*. The BFE device also needs to handle situations when the sensors connected to the BFE device malfunction or when they get disconnected from a patient. For example, for an ECG BFE device, the number of electrodes connected to a patient may vary from three to ten. The BFE device needs to continuously monitor the impedance of each electrode and generate an alert in case the electrode gets disconnected from the patient. The BFE device sends the sensor status to a PC periodically, and the PC can generate an alert in case of malfunction. Apart from generating an alert it may also be necessary to stop displaying the waveform of the disconnected ECG lead to avoid the doctor misinterpreting the noise picked up by the disconnected electrode as a valid ECG signal. The ECG BFE device may, however, continue to monitor the ECG signals from other electrodes.

  In another case, an air leak may be present in the cuff tied to the patient's arm for blood pressure measurement. The NiBP BFE device needs to monitor the rate of increase in cuff pressure once the air pump starts pumping air into the cuff. If the cuff pressure does not increase at the required rate an air-leak alert is transmitted to the PC.

- *Communication failure and communication errors*. The BFE device also needs to handle situations when the sensors and its hardware components are functioning normally but when the communication link to the AGG device or the PC fails. In such cases the BFE device is neither able to send the physiological data and its status to the PC, nor is it able to receive autonomic control messages from the PC. In such a scenario, the BFE device may save the patient's data in its local memory for transmission to the PC when the communication link is re-established. If the BFE device does not have sufficient memory to store data for long periods of disconnection, it may save the status of the last few minutes using a circular memory buffer. In case of communication failure, the BFE device keeps on looking for the periodic beacons sent by the AGG device in order to check the PAN-ID and re-establish the network connection.

  Communication errors may be introduced in the communication link between the BFE device, the AGG device and the PC due to the presence of other networks operating on the same wireless frequency. A typical example can be the interference of Wi-Fi (802.11b) in a Zigbee (802.15.4) network since they both operate in the same frequency band of 2.4 GHz. In another scenario, two patients may be connected with BFE devices that use the same frequency channel for communication to their respective AGG device. When such patients come in close proximity, their networks conflict with each other causing errors in communication. The BFE-AGG body area network needs to have a robust interference detection and mitigation algorithm by which a conflicting BAN would dynamically sense and change its operating frequency channel in case of interference from a similar network.

  In case the BAN is not able to mitigate the interference, robust error detection techniques should prevent the BFE device from executing wrong commands received from either the AGG device or the PC.

Self-Managing the Aggregator Device

The AGG device serves as a communication link between the WBAN formed by the BFE devices and the backend PC/server. The AGG device has a low-power wireless interface like 802.15.4 or a low-power Bluetooth to network the BFE devices and also a long range, high data rate interface like Wi-Fi to interface with the back-end PC. The AGG device may or may not have a local LCD display and control keypad. A typical example for an AGG device could be a PDA. The self-managing capabilities of the AGG device may be similar to that of the BFE devices.

The AGG device manages the BAN and allows BFE devices to join or leave the BAN seamlessly. Automatic BFE device discovery, device type identification, and BAN networking are the responsibility of the AGG device. The AGG should have sufficient capabilities to manage the BAN efficiently even in the absence of the communication link to the PC and have sufficient memory for storage of patients' physiological and status data over extended periods of operation.

Another important function of the AGG device is data encryption/decryption for network security.

Autonomic Platform Management (APM) by the PC

The PC or server is the central hub of data collection from multiple WBANs of multiple patients. The PC aggregates data from multiple patients, processes the data, stores them in specific formats and also makes the data available to multiple remote terminals for display. The PC also makes the patients' data available to the doctor at any place and at any time by transmitting important patient data to hand-held devices like mobile phones and PDAs. The PC also has the capability to issue control commands to any BFE device in its network to change its operating parameters. The PC also manages the shared resources like printers, fax equipment, etc., which are connected to it.

Since the PC is the central hub of information aggregation from BANs, user inputs from GUIs and information from remote terminals, it is better equipped to make self-management decisions for the entire health-monitoring framework. Also, since the PC has tremendous computation capacity it is better suited to run complex decision trees to arrive at the best management decision.

A single PC can be used to cater to the processing needs of multiple patients in a ward of the hospital. Multiple wards of the hospital can be connected to a server to further aggregate data from the entire hospital and to send/receive messages to hand-held devices.

The self-managing capabilities of the PC/server, shown by the APM layer in Figure 3, can be as follows.

- *Network management.* The PC forms a network of the AGG devices and dynamically changes the network topology as patients get transported in and out of the wards for tests and operative procedures. The PC is also aware of the devices in the BAN of each AGG device. The PC enables automatic device discovery, device association and disassociation algorithms, and automatic device-to-patient binding. Each patient is associated with a unique PAN-ID, and all devices in the BAN of the same patient have the same PAN-ID. The PC binds the real-time data of each patient to the patient's respective backend database.

  Apart from managing the networking of AGG and BFE devices, the PC also manages the shared resources on its network like printers, scanners, fax machines, etc.

- *BFE device control.* Since critical patients have different monitoring requirements as compared to non-critical patients in recovery, the PC can re-configure the BFE devices in different operating modes and change the device settings depending on the level of monitoring required.

  The PC is better equipped to handle self-management functionality of the health-monitoring platform as a whole. The PC may run complex bio-signal processing algorithms and based on the results may instruct the BFE device to execute certain self-management tasks. For example, the PC may analyze the ECG signal for the presence of baseline drift and may send a message to the ECG BFE device to restore the ECG baseline to ground by re-charging an AC coupling capacitor in the BFE device hardware. Similarly, if very small amplitude ECG signals are being sensed by the electrodes, it can instruct the BFE device to increase the amplification of the BFE device's programmable gain amplifier hardware.

  For a NiBP BFE device the PC may instruct the device to take blood pressure measurements either at programmed intervals or autonomically when certain thresholds are exceeded or under manual control.

  The PC can also perform autonomic power management of the devices connected on its network. The PC can turn ON or turn OFF the BFE devices or keep them in SLEEP state. The PC can wake the sleeping devices at appropriate times, instruct them to take a measurement and put them to SLEEP again. Also, the PC can extend the battery life of a device by dynamically changing the device parameters and settings. For example, the PC can instruct an ECG BFE device to reduce the number of leads to be sampled or reduce the sampling rate per lead, depending on the quality of ECG required.

- *Autonomic GUIs*. The PC can autonomically adjust the screen space to display a number of vital signal waveforms and numerical data from a number of patients. The screen can be automatically partitioned and re-adjusted to accommodate the data from new devices that join the network, and the screen space can be made free and utilized in a better manner when the BFE devices leave the network. When the screen space is limited, and displaying all parameters from all patients is not possible, the PC should give weight to critical parameters such as ECG readings.

- *Managing alerts*. The PC is the central computing resource for the entire health-monitoring framework and needs to process alerts coming from almost all devices attached to the network.

  Broadly, the alerts can be classified as alerts generated by the patient's physiological status, device status, and communication errors.

- *Alerts due to patient's physiological status*. The PC processes the physiological signals and parameter data from the patient and checks if any of the

programmed limits are exceeded. For example, the PC computes the heart rate by identifying and counting the number of 'R' waves in the patient's ECG per minute. In case the heart rate crosses the programmed limits, the PC may request the NiBP BFE device to take frequent blood pressure measurements and may also power ON a defibrillator machine in case the patient needs to be administered an electrical shock to restore normal heart rhythm. The PC can also intelligently identify irregular heart rhythms called arrhythmias and keep a log of these with a time stamp. In some critical conditions, the PC can automatically send the abnormal physiological data to a doctor's mobile phone/PDA for diagnosis and also trigger audio-visual alarms locally.

- *Device status alerts*. The PC monitors the status of the BFE and AGG devices and manages the alerts generated due to low battery, sensor failure or disconnection, component failure, calibration errors, etc. For example, if the patient condition is not critical, under low battery conditions the PC may instruct an SpO2 BFE device to intermittently monitor the blood oxygen saturation at 1-minute intervals, rather than monitoring it continuously. This allows the LEDs inside the SpO2 finger probe and the associated circuitry to sleep between measurements, thereby extending the battery life. For sensor disconnection the PC can raise appropriate audio-visual alarms and/or page the hospital staff on duty.

  In the case of component failure or calibration errors the PC can compensate for the error by software correction or instruct a device like the NiBP to measure blood pressure using the redundant pressure sensor. Once an error is detected and flagged by the PC, the PC also monitors whether the error has been corrected and removes the error flag accordingly. Alternatively, the user may manually acknowledge the error and put the PC in a mute state for a certain amount of time within which the user is expected to correct the error.

- *Fail-safe mechanisms*. Since the PC is the central computing resource for the health-monitoring platform, it is necessary to build reliability into the PC platform. Reliable hardware design, a reliable low-latency hard real-time OS, a reliable networking stack and application software are necessary to build a medical-grade PC. The PC should be immune to the EMI generated by other medical equipment in its vicinity and should also have low EMI emissions. The PC should have a battery backup to remain operative in case the mains power supply fails. The PC should have the capability of self-monitoring, preventive maintenance, and have redundant processing cores and configurable logic to heal itself in case of component failure. The storage media and the Internet broadband link should be robust. The aggregated patient data and the network status information should also be copied in a central archive so that in case of malfunction, a standby PC should be able to take over the monitoring functionality of the failed PC without loss of data.

USAGE MODELS

This self-managed digital health-monitoring framework can be applied in several use cases, some of which are described below:

Use Case 1: Remotely Monitoring Patients at Home

Home monitoring involves monitoring of non-critical patients, patients in the recovery stage after being discharged from hospital, proactive health monitoring, or monitoring the health of the elderly within their homes or care facilities.

Joe, 68 years old, has recovered from open heart surgery and has been discharged from the hospital. Lately, Joe has become forgetful and forgets even basic tasks like taking his medicines on time. Joe's children are working and are concerned about Joe's health when they are at work. Dr. Smith wants to remotely monitor Joe's recovery as Joe performs his daily activities to ensure that his recovering cardiovascular system is not subject to a sudden stress. Joe is fitted with a wireless ECG device, a NiBP device, and a Pulse Oximetry device for monitoring his blood oxygen saturation. The devices connected to Joe are miniature, lightweight, and they do not interfere with his daily activities.

Joe has a Wi-Fi-enabled desktop PC in his living room which is also used by Joe to watch TV. The PC continuously monitors Joe's vital parameters in the background and sends data to the hospital in real-time. Wireless webcams fitted in different rooms of his home also send streaming video data to the PC. Joe's PC is connected to the Internet using a broadband link.

While sitting at his laptop in the hospital, Dr. Smith connects to Joe's PC and runs the pre-designed protocol for remotely monitoring open heart surgery patients like Joe. Dr. Smith runs the protocol and then leaves for an urgent operation. The protocol remotely and autonomically programs Joe's ECG device to continuously monitor a 3-lead ECG, the NiBP device to take a blood pressure reading once every hour, and the Oximetry device to monitor only in critical situations. The protocol automatically sends a reminder audio-visual message on Joe's PC screen asking him to undergo the

morning exercise on his home treadmill and also programs the treadmill to limit the maximum speed to 2 miles per hour and limit the exercise time to 10 minutes. The reminder message alerts Joe in the midst of his TV program. Joe starts walking on the treadmill and during his exercise session, his heart rate increases to 130 beats per minute (bpm). The increased heart rate is monitored by the PC, and the PC triggers the NiBP device to take frequent blood pressure measurements and it also automatically stops the treadmill as a precautionary measure to prevent excessive stress. The increased heart rate condition also turns on the Pulse Oximetry device to monitor Joe's oxygen saturation. When Joe's heart rate returns to normal after his exercise session, the NiBP device again increases the BP measurement interval to 1 hour while the Oximetry device turns OFF. Since Dr. Smith had set the mobile phone alert limit for heart rate to 150 bpm, BP limits to 150/100 mmHg, and the SpO2 limit to 94%, he was not alerted on his mobile phone since Joe's vital signs were within the limits. When Dr. Smith returns from his operation he examines the stored vital parameters of Joe and he is happy with Joe's increased stress-handling capability. Meanwhile, Joe's son is also able to remotely keep an eye on Joe from his office by monitoring the streaming webcams and Joe's vital parameters. Dr. Smith has remotely also fed the medicine schedule on Joe's PC. Joe's PC flashes audio-visual messages on its screen in a timely manner to remind Joe to take his medicines.

Use Case 2: Wireless Fetal Monitoring During Labor

Judy is pregnant and has been admitted to the hospital for delivery. Dr. Willy expects a normal delivery but decides to monitor Judy's Uterine Contractions (UC) and Fetal Heart Rate (FHR). The fetal heat rate variability is an important parameter to asses fetal well being and to assess whether the fetus would be able to sustain the stress of uterine contractions during delivery. Dr. Willy fits Judy with a wireless ultrasound sensor and a uterine pressure sensor by means of a belt around her abdomen. The ultrasound sensor monitors the FHR while the pressure sensor monitors Judy's UCs and sends the data wirelessly to the central server of the labor ward, where a number of patients are being monitored simultaneously. Judy is still not in labor and is able to take frequent walks around her room and in the ward, while still being closely monitored. Judy does not feel the subtle UCs, indicating the start of labor. However, minute uterine pressure changes are picked up by the pressure sensor and are evident in the graphical tracing at the central monitoring terminal. There is a sudden and steep drop in the FHR which goes unnoticed by the busy hospital staff. However, the FHR analysis software on the server detects the sudden drop in FHR, and the software automatically sends alert messages and FHR tracings to Dr. Willy and the chief nurse on their mobile phones. Dr. Willy is out of hospital when he receives the message on his mobile phone. Dr. Willy immediately rings up the chief nurse and instructs her to be prepared for an emergency cesarean operation by the time he reaches the hospital. Judy undergoes an emergency cesarean operation and delivers a healthy baby. During delivery Dr. Willy notices the umbilical cord entangled around the fetus's neck resulting in partial suffocation and thereby decreasing the FHR. Dr. Willy is thankful that the timely alerts by the central monitoring system saved the baby's life.

Use Case 3: Critical Patient Monitoring in an Intensive Care Unit

Dr. Bill is a resident doctor currently managing the 16-bed ICU in a hospital. A patient, Jack, operated on via angioplasty, is brought to Dr. Bill's ICU. Dr. Bill tags Jack's admission time in ICU by using a RFID reader to read Jack's RFID wrist strap and also wirelessly transfers a softcopy of Jack's medical record file from inside the wrist strap to the central station for study. Dr. Bill rubs the wireless patient monitoring devices on Jack's RFID wrist strap and connects them to his body. The wireless monitoring devices read Jack's Patient-ID from his wrist-strap to form a BAN and get associated to Jack's backend database on the central station. The computer screen of the ICU's central monitoring station automatically re-adjusts itself to display, monitor, and store Jack's vital signs along with the parameters of other patients. Dr. Bill prepares a monitoring schedule and a drug administration schedule for Jack using the central monitoring application software. The monitoring and drug administration profile for Jack is immediately transmitted to the hand-held devices given to nurses on duty in the ICU. The central station performs periodic diagnostic tests using the attached wireless devices and also sends timely reminders to the nurses to administer drugs to patients, depending on individual patient profiles. Once a nurse administers a drug to a patient she mutes the generated reminder and logs the event, which also gets logged in the central station. The central monitor also runs intelligent algorithms on patient's vital physiological signals and generates alerts, sends alerts to doctors' mobile phones, and prints important events on locally connected printers. Dr. Bill is relieved that the central monitoring station manages most of his mundane tasks and helps him in effectively managing large numbers of patients.

CHALLENGES AND OPPORTUNITIES

There are several challenges which need to be overcome in order to realize such a digital health-monitoring framework:

Platform reliability. Ensuring the reliability of the entire platform is the key to the success of the proposed architecture. Reliability has to be built right from the electronic component level to the OS and application software level. It is necessary to have fail-safe and backup mechanisms to ensure that patient monitoring is not interrupted when parts of the network fail.

Robust wireless communication. Wireless communication is the backbone of the proposed health-monitoring framework. Robust mechanisms should be developed to mitigate interference issues that result when several wireless networks co-exist. Robust error correction and error detection algorithms should be developed to build the same reliability as that of a wired link in the wireless interfaces.

Standardization and interoperability. The medical device OEMs need to agree on a common set of communication protocols and standard hardware interfaces. Devices from different manufacturers should be able to plug into the system seamlessly.

Infrastructure and ubiquity. The health-monitoring framework needs wireless communication infrastructure like Wi-Fi hot spots, routers, switches, etc. which are limited to a hospital or home network. Emerging technologies such as WiMAX can address the problem of seamless wireless connectivity throughout the cities and villages.

Intelligent algorithms. The PC needs to run intelligent algorithms to make self-managing decisions. The algorithms should be continuously evolving and patient centric. Machine learning, artificial intelligence, and prediction algorithms may require tweaking and clinical trials until they reliably self-manage a health-monitoring framework.

Development of miniature, ultra-low power sensors and battery technology. BFE device sensors and hardware need further miniaturization to the level of a small system-on-chip, and the power needs to be optimized to the microwatt level so that the sensors can operate by using ambient light as the power source. Battery technology such as the lithium-polymer and moldable lithylene battery technology have to evolve further to combine high capacity, small form factor, and light weight.

CONCLUSION

We believe that a self-managed wireless health-monitoring framework can significantly improve the quality of healthcare while providing patient comfort, mobility, and continuity of care. Such a framework does not require the doctor to be in close proximity to the patient; however, it still provides the same quality of care. The doctor is enabled to monitor more patients effectively. The number of errors related to paper-based processes is reduced significantly in an autonomically managed framework. The proposed framework leads to better resource utilization, better resource sharing, reduces doctor's intervention, and hence makes healthcare more affordable.

Such an open standards-based health-monitoring platform would motivate more standards-based hardware and software designs and shift the biomedical OEMs from proprietary hardware-centric platforms to standards-based software-centric general-purpose PC platforms.

ACKNOWLEDGMENTS

The authors recognize the valuable suggestions given by the reviewers for improving the quality and content of this paper.

REFERENCES

[1] David Culler, Deborah Estrin, Mani Shrivastava, "Overview of Sensor Networks," *IEEE Computer Society*, August 2004, pp. 41–49.

[2] J. A. Stankovic, Q. Cao, "Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges," *Department of Computer Science, University of Virginia.*

[3] Chee-Yee Chong, Srikanta P. Kumar, "Sensor Networks: Evolution, Opportunities and Challenges," in *Proceedings of the IEEE*, Vol. 91, No. 8, August 2003, pp. 1247–1256.

AUTHORS' BIOGRAPHIES

Amit Baxi is a biomedical engineer working as an R&D Engineer in the Corporate Technology Group, Intel Corporation. He has been in Intel for more than a year and is responsible for architecting and building the hardware and software bio-medical components for Intel's healthcare platforms. He has more than 11 years experience in the design and development of medical embedded systems such as Cardiac Stress Test Systems, Defibrillators, Multiparameter Patient Monitoring Systems, Spirometers, ECG monitoring and diagnostic equipment etc. His e-mail is amit.s.baxi at intel.com.

Nagaraju Kodalapura is an electronics and communication engineer working as an R&D Engineer in the Health Platforms lab of the Corporate Technology Group, Intel Corporation. He has been with Intel for about six years. He is primarily responsible for the design and development of Wireless Embedded System Software for bio medical sensors for Intel's healthcare platforms. His expertise is in the area of wireless embedded systems, device drivers, debuggers, and simulators for multicore

processors. His e-mail is nagaraju.n.kodalapura at intel.com.

THIS PAGE INTENTIONALLY LEFT BLANK

For further information visit:

developer.intel.com/technology/itj/index.htm