A REAL-TIME WEB-BASED HEALTH MONITORING SYSTEM BASED ON ENTERPRISE SERVICE BUS

Yohanes Baptista Dafferianto Trinugroho, Kamyar Rasta, Trinh Hoang Nguyen, Rune Fensli and Frank Reichert
Faculty of Engineering and Science, University of Agder
Jon Lilletuns vei 9, 4879 Grimstad, Norway

ABSTRACT

Remote health monitoring is becoming more prevalent in recent years, mainly to support deployments of telehealth services run by health service providers. Telehealth is one of the most promising solutions to tackle geographical issues related to healthcare provisioning and to improve timely dissemination of health-related information from distantly-located patients to healthcare personnel. In addition, telehealth can alleviate the cost burden of healthcare services by moving non-urgent treatments from healthcare premises to patients' homes. Patients of chronic diseases in particular and elderly people in general may require real-time monitoring of their health status wherever they are. This is currently feasible to achieve by employing portable on-body wireless sensors for reading vital signs within Wireless Body Area Networks, and Internet technologies. These information is then forwarded to a remote health monitoring server and responsible personnel (doctors, nurses, family members etc.) are notified in a real-time manner. This paper presents an architecture for a real-time web-based health monitoring system by utilising Enterprise Service Bus for centralised data integration from different data sources. An early prototype of the implementation result is also presented.

KEYWORDS

Health monitoring, Enterprise Service Bus, SOA, publish/subscribe, real-time, web.

1. INTRODUCTION

Improvements in healthcare quality and decreasing number of birthrate have contributed to the increasing number of elderly population in many developed countries. Statistics Norway, for example, reported that around 625,000 people older than 67 years old lived in Norway in 2010, and the figure is expected to double in 2060 (Mørk, 2011). This situation, however, is not balanced with the increasing number of healthcare workforce, which widens the gap between demand and supply within healthcare sector (Sataline and Wang, 2010). On the other hand, healthcare cost has been steadily increasing each year. In the United States, the total healthcare expenditure was around $2.6 trillion in 2010 alone, and the amount is expected to double within five years (Fife and Pereira, 2011). Many have argued that early detection and preventive care are solutions to lower the cost pressure of healthcare services. Telehealth provides capabilities to assist health maintenance and detection by utilising information and communications technology, eliminating distance barrier between healthcare provider and patient (Gagnon et al., 2011). In addition, telehealth supports remote alarming services as well which enables healthcare personnel to be notified whenever emergency situation occurred to the distantly-located patient.

Remote health monitoring is one of the most important applications of telehealth as it provides measurements and reports of patients' activities and conditions to distantly-located healthcare providers. It enables healthcare personnel (e.g. doctors, nurses) as well as family members and relatives to remotely monitor elderly people and patients with special needs in their daily activities. This will provide the patients with more independence in living their lives with minimum physical intervention from others. Remote health monitoring is made possible by utilising different devices/sensors in the patient's surroundings, usually deployed in a smart home environment. Most of these devices are worn by the patient, especially the ones which measure vital information. However, assuming a patient to always stay at home is not very realistic, especially for those with good mobility condition. On the other hand, promoting elderly people to have a

ISBN: 978-989-8533-09-8 © 2012 IADIS

short walk for a couple of hours a day may give positive impact to their health. Thus, the health monitoring system should also enable patient monitoring in outdoor environments. Advancements in wireless communications technology have enabled reduction of wires in devices/sensors around the patient's body, forming so-called Wireless Body Area Networks (WBAN). Although the measured vital signs information may not be as thorough as in smart home environment, the outdoor health monitoring system is at least able to inform responsible personnel the patient's latest health condition. The pervasiveness of Internet connectivity in recent years has enabled such information to be transmitted to a centralised monitoring server in a real-time manner.

This paper presents an architecture for real-time web-based health monitoring system which uses Enterprise Service Bus as messaging backbone in the health monitoring server to support a publish/subscribe messaging model. Early implementation results are also presented.

2. RELATED WORK

There are some related previous works within the area of remote health monitoring which are interesting to consider. Authors in (Laguna et al., 2009) proposed a product line generic architecture with an example application of real-time health monitoring using a wireless sensor connected to a central station by means of a smart phone, employing Service-Oriented Architecture (SOA) paradigm. However, possibility for message flow control from different data sources (e.g. sensors) is not highlighted. A SOA framework for WBAN-based patient monitoring was proposed in (Abousharkh and Mouftah, 2011), where sensors are coordinated by a node which is responsible to retransmit the signals to a remote central monitoring unit. The proposed middleware makes use of web services, but how the messages being handled within the back-end monitoring server to the client side terminal is not described. Authors in (Hongzhou and Lu, 2011) described a remote health monitoring system by employing ZigBee protocol to transfer all data output from medical devices to a GPRS gateway, which then forwards them to healthcare center for further analysis. How these data is being reported to healthcare personnel is not presented in detail as the main focus is the usage of ZigBee protocol for gathering the data from sensors. A smartphone-based healthcare system providing real-time continuous monitoring of health conditions was proposed in (Nawka et al., 2011). Several sensors are included in the implementation, but how healthcare personnel are being notified is not clearly described.

3. GENERAL SYSTEM DESIGN

Real-time health monitoring system requires measurements of vital signs from sensors worn by the patient to be sent to a back-end monitoring server for notification and further analysis. Each sensor can send the measurements directly to the back-end monitoring server or through a gateway which then relays the aggregated information. A direct approach, which is shown in Figure 1a, is not very feasible to be adopted in outdoor environments where Internet connectivity is mainly provided through cellular links. This will require each sensor to have its own SIM card, and additional sink servers to receive each sensor's data before being aggregated by the monitoring server. Therefore, a gateway-based approach is more suitable for a deployment of "anytime, anywhere" remote health monitoring system, as shown in Figure 1b.

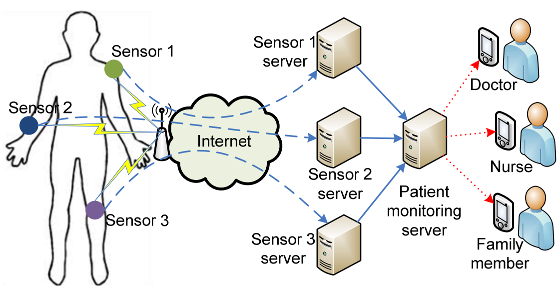

Figure 1a. Direct data transmission

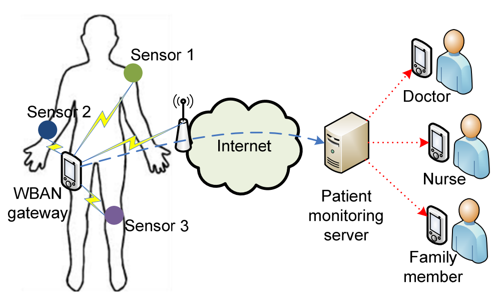

Figure 1b. Brokered data transmission

As smartphones' processing capabilities have increased quite drastically within the last couple of years nearing to the performance of personal computers, coupled with ever decreasing price tags, it is expected that smartphones will become the standard communications devices in the near future. New generation of elderly people is expected to use smartphones in their daily lives as well. From this standpoint, treating smartphones as WBAN gateways for remote health monitoring purposes is reasonable, as shown in Figure 1b. The smartphone carried by the patient will aggregate data from different sensors within the WBAN (e.g. via Bluetooth) and transmits them to the back-end monitoring server.

Transmitting all gathered data from sensors to the monitoring server may not be necessary as many of the measurements' values may be similar or are only slightly different from previous readings. In this case, it is useful if the smartphone can do a sort of reasoning every time new data from a sensor arrives. This will minimise the Internet bandwidth used between the smartphone and the monitoring server. To achieve this, a context-aware application agent should be deployed in the smartphone. This application agent can be used to detect anomalies of the patient's health status and sends an alarm message to the monitoring server as well as directly contacting responsible healthcare personnel. However, battery conservation needs to be taken into account as well, and heavy reasoning processes in the smartphone may have negative effects. Therefore, more sophisticated reasoning processes are proposed to be employed at the monitoring server. Figure 2 illustrates the general idea of 2-level reasoning processes. Details of the reasoning processes are out of scope of this paper.

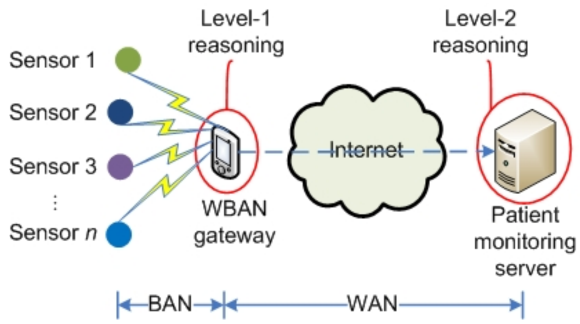

Figure 2. 2-level of reasoning processes

4. ENTERPRISE SERVICE BUS AS MESSAGING BACKBONE

SOA has become a major trend in the last couple of years as a way to support business processes of organisations, especially using web services technology (Barry, 2003). SOA promotes interoperability between different software applications running on disparate systems by employing open standards and protocols, separating implementation logic and interface of a service. Combining SOA paradigm with event-driven processing lays foundation for an emerging technology that amalgamates various conventional distributed computing, middleware, Business Process Modelling (BPM) and Enterprise Application Integration (EAI) technologies. It offers a unified backbone on top of which enterprise services can be advertised, composed, planned, executed, monitored, and decommissioned (Papazoglou and Van Den Heuvel, 2007). This messaging implementation backbone is referred to as Enterprise Service Bus (ESB). It acts as a loosely coupled, event-driven SOA with a highly distributed universe of named routing destinations across a multi-protocol message bus. Unlike traditional web services point-to-point approach which could not avoid tight-coupling between service consumer and service producer of the messages being exchanged, ESB provides a centralised approach for integration tasks, avoiding direct contacts between communicating services. Applications in ESB are abstractly decoupled from each other, and connected together through the bus as logical endpoints that are exposed as event-driven services. In general, an ESB has four major functions: message routing, message transformation, protocol mediation, and event handling (Chappell, 2004). ESB supports various message exchange patterns, and one of the most important ones is publish/subscribe model, where applications can subscribe to a specific topic and notified whenever a data provider for that particular topic publishes a message for that topic.

ISBN: 978-989-8533-09-8 © 2012 IADIS

Within healthcare arena, ESB is mainly used to integrate different health information systems from various healthcare providers and vendors. In this paper we propose ESB to be utilised as messaging backbone for remote health monitoring, acting as a single messaging hub for integration of all information sent from the smartphone at the patient side, and deployed in the monitoring server. The event-driven nature of ESB is taken advantage of in our system to support real-time message dissemination by incorporating the publish/subscribe message exchange pattern between data providers (i.e. WBAN sensors) and client application (i.e. remote health monitoring dashboard).

5. IMPLEMENTATION RESULTS

Initial prototype of the remote health monitoring system has been implemented. The system architecture of the proposed system is depicted in Figure 3. As web services have become the de facto standard for interoperable machine-to-machine interaction, web services interfaces are provided in the system for data input. The on-body sensors worn by the patient become data providers, with smartphone gateway as a relay. The smartphone application agent performs light reasoning processes before retransmitting the gathered information from sensors such as checking whether the new data value from a particular sensor is equal to the previous value. Web-based client application is chosen instead of native application for a specific platform since web-based application is more pervasive in today's information technology landscape as web browsers are generally available for all platforms. This will avoid installation needs for specific platforms.

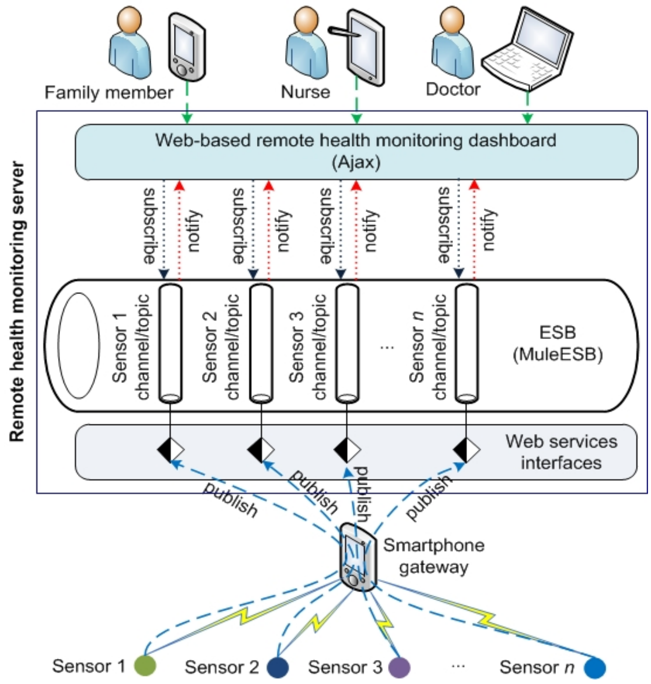

Figure 3. System architecture

5.1 Smartphone Application

For the prototype implementation, an Android-based smartphone is used as the WBAN gateway for different sensors. The sensors have not been integrated yet, and in order to demonstrate the system, different values are generated within the normal range for pulse rate, body temperature, and blood oxygen saturation. These values are then checked by the application whether they are similar to the previous values. If they are exactly the same, those data will not be transmitted to the monitoring server. This will minimise Internet bandwidth usage, which in turn will save some cost. Since the remote health monitoring system is aimed to be used in both indoor and outdoor environments, location information is of high importance, especially for the latter case. Location information can be used to locate the patient if an emergency situation occurs as well as track their travelling route for further analysis. Internal GPS receiver of the smartphone is utilised for this purpose by gathering the latitude and longitude information of the smartphone. The current configuration for the location listener is set to receive notifications of new location from both *GPS_PROVIDER* and *NETWORK_PROVIDER* (WiFi and 3G), which are made available through the *LocationManager* class, only if the patient moves 10 meters from the last-known location, and the accuracy should be less or equal than 20 meters for outdoor GPS location provider and less or equal than 60 meters for indoor WiFi network provider. The average accuracy during several testing sessions is 6 meters for outdoor GPS location provider and 50 meters for indoor WiFi network provider. The application presents the patient with his/her location on a map as well as the measurements values.

The application makes use of *ksoap2* web services client library to wrap the gathered data as SOAP messages and send them to the back-end server through the web services end-points. Figure 4 shows a screenshot of the smartphone application.

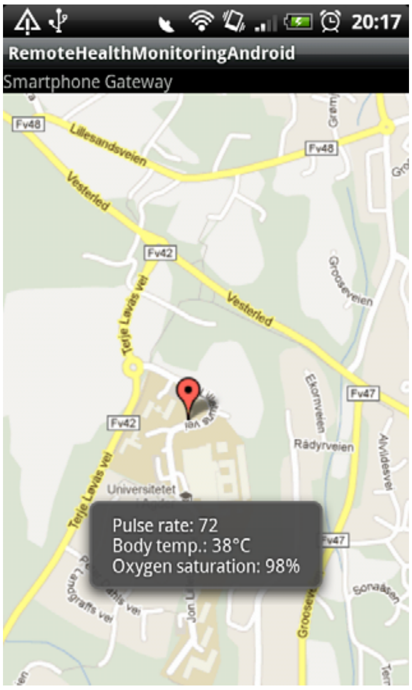

Figure 4. Smartphone application

5.2 Remote Health Monitoring Server Application

The server-side application consists of the ESB and a web-based client application based on *Ajax*. For the initial prototype implementation, open source *MuleESB* software is used due to its wide range of transport protocols support as well as ease of development by means of friendly integrated development environment called *MuleStudio*. The web-based client application provides a remote health monitoring dashboard to healthcare personnel as well as family members that can be accessed pervasively using various web browsers of choice (including those on mobile devices) through the Internet. This client application subscribes to different topics/channels of sensor information fed by the smartphone application at the patient side. Whenever new information comes to any of the subscribed channel/topic, the client *Ajax* application is notified by means of callback functions. By utilising the publish/subscribe message exchange model which is supported by the ESB, a real-time remote health monitoring dashboard can be provisioned. In the initial prototype, four channels/topics are deployed for four different information: location, pulse rate, body temperature, and blood oxygen saturation.

MuleESB can be used to host different *Mule* applications. A *Mule* application is a deployment unit that encapsulates the necessary requirements which an application would need to function, such as libraries, custom code, and any environment settings accompanying the application (Perepelytsya and Magnusson, 2011). The *Mule* application contains one or more *Mule* flows in the form of XML configuration files. A *Mule* flow consists of pre-packaged building blocks which are defined in specific sequences. *Mule* application processes and orchestrates *Mule* messages based on those sequences (Bock and Dickey, 2012). *MuleStudio* provides two ways to modify building blocks in a *Mule* flow either through the graphical user interface or writing corresponding XML tags of those building blocks directly in the XML configuration file.

A *Mule* message crosses from one block to the next block in the *Mule* configuration file while each block processes the message and takes actions according to its configurations. A *Mule* message consists of three parts: header, payload, and an attachment. The header contains sets of properties and is used to route the message to the destination. The payload consists of business-specific data such as body temperature that is read from a sensor. Each message can also carry an optional header along with it.

Ajax is a set of interrelated technologies that simplify the creation of asynchronous web applications. With the aid of *Ajax*, web applications are able to send and receive data to and from servers while maintaining the view of existing page unchanged (Ullman and Dykes, 2007). *MuleESB* provides *Ajax* namespace and *Ajax* connector in order to bind *Mule* flow and web services to the *Ajax* channel on the browser. The *Ajax* endpoint is configured as an outbound operation. It creates a transport channel to send messages asynchronously to and from an *Ajax* server, which connects *Mule* flow to an external web page. A JavaScript function attached to the web-based client application listens for incoming messages. It extracts and classifies the received data and shows it on the web page.

Figure 5a, 5b, 5c, and 5d depict mappings of *MuleStudio* flows to their corresponding XML configurations in *MuleESB* for handling the aforementioned four different information.


```
<flow name="LocationFlow" doc:name="LocationFlow">
    <inbound-endpoint address="http://128.39.201.109:65082/services/Location" doc:name="Location Inbound"/>
    <cxf:jaxws-service port="80" serviceClass="no.uia.ehealth.ws.Location" enableMuleSoapHeaders="false" doc:name="SOAP"/>
    <component doc:name="Java">
        <singleton-object class="no.uia.ehealth.ws.Location"/>
    </component>
    <ajax:outbound-endpoint channel="/services/location" connector-ref="AjaxConnector" doc:name="Ajax"/>
    <echo-component doc:name="Echo"/>
</flow>
```

Figure 5a. Location information flow


```
<flow name="PulseRateFlow" doc:name="PulseRateFlow">
    <inbound-endpoint address="http://128.39.201.109:65082/services/PulseRate" doc:name="PulseRate Inbound"/>
    <cxf:jaxws-service port="80" serviceClass="no.uia.ehealth.ws.PulseRate" enableMuleSoapHeaders="false" doc:name="SOAP"/>
    <component doc:name="Java">
        <singleton-object class="no.uia.ehealth.ws.PulseRate"/>
    </component>
    <ajax:outbound-endpoint channel="/services/pulserate" connector-ref="AjaxConnector" doc:name="Ajax"/>
    <echo-component doc:name="Echo"/>
</flow>
```

Figure 5b. Pulse rate information flow


```xml
<flow name="BodyTempFlow" doc:name="BodyTempFlow">
    <inbound-endpoint address="http://128.39.201.109:65082/services/BodyTemp" doc:name="BodyTemp Inbound"/>
    <cxf:jaxws-service port="80" serviceClass="no.uia.ehealth.ws.BodyTemp" enableMuleSoapHeaders="false" doc:name="SOAP"/>
    <component doc:name="Java">
        <singleton-object class="no.uia.ehealth.ws.BodyTemp"/>
    </component>
    <ajax:outbound-endpoint channel="/services/bodytemp" connector-ref="AjaxConnector" doc:name="Ajax"/>
    <echo-component doc:name="Echo"/>
</flow>
```

Figure 5c. Body temperature information flow


```xml
<flow name="OxygenSaturationFlow" doc:name="OxygenSaturationFlow">
    <inbound-endpoint address="http://128.39.201.109:65082/services/OxygenSaturation" doc:name="OxygenSaturation Inbound"/>
    <cxf:jaxws-service port="80" serviceClass="no.uia.ehealth.ws.OxygenSaturation" enableMuleSoapHeaders="false" doc:name="SOAP"/>
    <component doc:name="Java">
        <singleton-object class="no.uia.ehealth.ws.OxygenSaturation"/>
    </component>
    <ajax:outbound-endpoint channel="/services/oxygensaturation" connector-ref="AjaxConnector" doc:name="Ajax"/>
    <echo-component doc:name="Echo"/>
  </flow>
```

Figure 5d. Blood oxygen saturation information flow

Sophisticated reasoning has not been implemented yet at the server-side in this initial prototype. Java components can be used for reasoning purposes in the message flow in future extensions. Figure 6 depicts the remote health monitoring dashboard accessed by a web browser. The location of the patient is rendered in real-time on the browser's canvas whenever a new latitude and longitude information is sent by the smartphone application. Any new information from the sensors is also updated instantaneously.

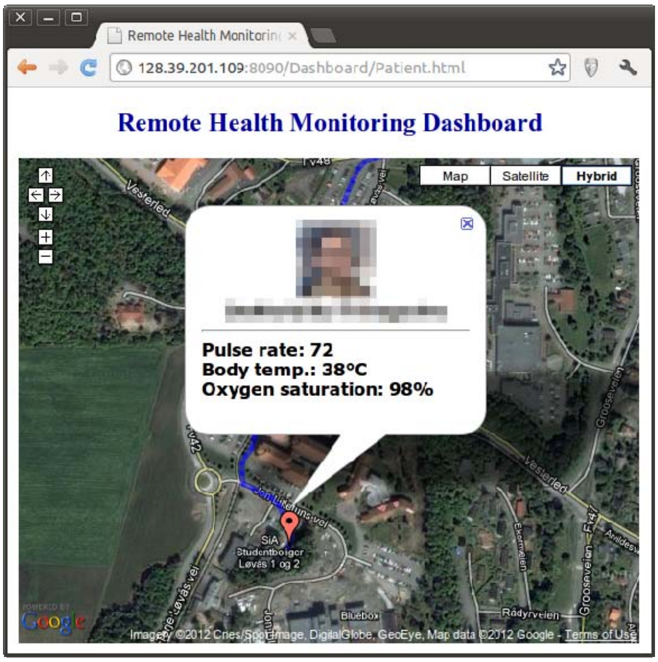

Figure 6. Web-based remote health monitoring dashboard

6. CONCLUSIONS AND FUTURE WORK

Telehealth has been widely used in recent years to provide healthcare services at a distance. It supports eliminating space barriers and improves access to healthcare services. Remote health monitoring is an important part of telehealth which enables early detection of health anomalies and preventive distant care. A web-based remote health monitoring "anytime, anywhere" system based on ESB is presented in this paper, enabling distantly-located healthcare personnel and family members of the patient to monitor the patient in a real-time manner with a web browser of choice. This is achieved by employing publish/subscribe message exchange pattern functionality of the ESB. Web services interfaces are used to expose the inbound data entry point to the monitoring flow within the ESB. A smartphone gateway is used at the patient side as data sink for on-body WBAN sensors worn by the patient. Simple reasoning processes are carried out by the smartphone application before transmitting the aggregated data to the monitoring server through the Internet.

Planned future work includes more sophisticated reasoning processes at the monitoring server side, incorporating ontologies and semantic technologies. Role-based access control mechanism is also envisaged to be included for the monitoring dashboard alongside with integration of real sensors.

REFERENCES

Abousharkh, M. and Mouftah, H., 2011. Service Oriented Architecture-based Framework for WBAN-enabled Patient Monitoring System. *Proceedings of the Second Kuwait Conference on e-Services and e-Systems.* Kuwait City, Kuwait, pp. 1-4.

Barry, D. K., 2003. *Web Services and Service-Oriented Architecture: The Savvy Manager's Guide*. Morgan Kaufmann Pub., San Francisco, CA, USA.

Bock, N. and Dickey, A., 2012. Mule Application Architecture - Mule ESB 3.3 User Guide. Available: http://www.mulesoft.org/documentation/display/MULE3USER/Mule+Application+Architecture [Accessed 24th July 2012].

Chappell, D. A., 2004. *Enterprise Service Bus*. O'reilly Media, Sebastopol, CA, USA.

Fife, E. and Pereira, F., 2011. Digital home health and mHealth: Prospects and Challenges for Adoption in the U.S. *Proceedings of 2011 50th FITCE Congress (FITCE).* Palermo, Italy, pp. 1-11.

Gagnon, M. et al, 2011. Supporting Work Practices through Telehealth: Impact on Nurses in Peripheral Regions. *BMC Health Services Research,* Vol. 11, No. 1, pp. 1-9.

Hongzhou, Y. and Lu, L., 2011. Remote Health Monitoring System Using ZigBee Network and GPRS Transmission Technology. *Proceedings of 2011 Fourth International Symposium on Computational Intelligence and Design (ISCID)*. Hangzhou, Zheijiang, China, pp. 151-154.

Laguna, M. et al, 2009. Remote Health Monitoring: A Customizable Product Line Approach. *Lecture Notes in Computer Science,* Vol. 5518, pp. 727-734.

Mørk, E., 2011. Seniorer i Norge 2010. Available: http://www.ssb.no/emner/00/02/sa_seniorer/sa120/sa120.pdf [Accessed 20th March 2012].

Nawka, N. et al, 2011. SESGARH: A Scalable Extensible Smart-Phone based Mobile Gateway and Application for Remote Health Monitoring. *Proceedings of 2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA).* Bangalore, India, pp. 1-6.

Papazoglou, M. P. and Van Den Heuvel, W. J., 2007. Service Oriented Architectures: Approaches, Technologies and Research Issues. *The VLDB journal,* Vol. 16, No. 3, pp. 389-415.

Perepelytsya, A. and Magnusson, K., 2011. Application Format - Mule ESB 3.3 User Guide. Available: http://www.mulesoft.org/documentation/display/MULE3USER/Application+Format [Accessed 24th July 2012].

Sataline, S. and Wang, S. S., 2010. Medical Schools Can't Keep Up. Available: http://online.wsj.com/article/SB10001424052702304506904575180331528424238.html [Accessed 10th May 2012].

Ullman, C. and Dykes, L., 2007. *Beginning Ajax*. Wrox Press.