Design, Simulate and Prototype Data Decision System for the Smart Universal Gateway for e-HealthCare System

Boidi Krishna Verma

THESIS WORK 2011
Electrical Engineering
Design, Simulate and Prototype Data Decision System for the Smart Universal Gateway for e-HealthCare System

Boidi Krishna Verma

This thesis work is performed at School of Engineering, Jönköping University within the subject area Electrical Engineering. The work is part of the master's degree programme with the specialization in Embedded Systems. The author is responsible for the given opinions, conclusions and results.

Supervisor: Youzhi Xu

Credit points: 30 points (D-level)

Date:

Archive number:
Acknowledgement

First of all, I am very much thankful and would like to express a lot of respect to my supervisor Professor Youzhi Xu, Jönköping School of engineering for encouraging and guiding me throughout the Master thesis. I am also very grateful for his support for providing required tools and equipment for undergoing master thesis, providing very good working environment at WSN Application Labs, Science Park. Under his supervision, I had a great opportunity to learn many things during very meeting sessions.

I would also like to thank Lab Coordinators Wei Shen and Hongwei Huo for being available all times, help me by providing valuable suggestions.

I would also express lot of respect and thanks to the Master Program Coordinator, senior Lecturer Alf Johansson for being always helpful and caring.

I would also like to thank my entire teachers for providing valuable education, guidance throughout studies and are of great knowledge.

Last but not least, I would like to express thanks, lots of respect and love to my parents Boidi Laxmi bai and late Boidi Narayana Verma. I am very grateful towards them for their guidance, support and motivation.
Abstract

With advances in low-power and low-cost wireless and sensor technologies, Wireless Sensor Networks (WSN) is playing an important role in the eHealth Care Sector. In such WSN applications, an intelligent System is needed to act as a network manager for WSNs and a communication bridge between WSNs and other networks for forwarding the health attributes of the elderly at home to the remote health care givers. In the thesis, a Smart Intelligent Universal Gateway with a Data Decision System is proposed. WSN nodes samples and transmits the health information of the elderly at home such as blood pressure, heartbeat, pulse rate etc to the Smart Universal Gateway. The final health states are decided by Smart Universal Gateway using the Data Decision System built within it. The manipulated results are derived in the form of the Alarms, the priority messages and the information Acknowledgements to the care givers including Health Care Centre.

A Hidden Markov Model is implemented to improve the decision process of our Data Decision System (DDS). We have implemented and evaluated the DDS on a real platform, ARM board.
Sammanfattning

Genom teknologiska framsteg inom låg effekt och låga tillverkningskostnader har trådlösa sensornätverk, WSN, börjat att spela en betydande roll inom e-hälsa (eHealth Care). I sådana WSN-tillämpningar krävs intelligenta system som kan fungera som ”Network Manager” och också vara en brygga mellan WSN och andra nätverk, vilka skickar de äldres hälsostatus vidare till vårdgivare.

I detta examensarbete föreslås en ”Smart Intelligent Universal Gateway” med beslutstöd ("Data Decision System, DDS"). WSN-noder samlar in data och skickar de äldres hälsostatus, t.ex. blodtryck och EKG, till ”Smart Intelligent Universal Gateway” där slutlig hälsostatus bestämmas med hjälp av DDS. Som resultat av detta skapas alarm och prioriterade meddelanden som skickas till vårdgivare.

För att förbättra beslutstödet använder DDS en ”Hidden Markov Model”. DDS har implementerats och utvärderats med hjälp av ett ARM-baserat datorkort.

Keywords

Universal Gateway
Data Decision System
Hidden Markov Model
Wireless Sensor Networks
Transfer Control Protocol /Internet Protocol
Network File System
Table of Contents

1 Introduction ............................................................................................................. 1
  1.1 BACKGROUND ................................................................................................. 1
  1.2 PURPOSE AND AIMS ....................................................................................... 2
      1.2.1 Requirements Functions of Smart Gateway: ............................................. 2
      1.2.2 Smart Universal Gateway Onboard-server functional requirements............. 3
  1.3 DELIMITS .......................................................................................................... 3
  1.4 OUTLINE .......................................................................................................... 3

2 Theoretical background ............................................................................................. 4
  2.1 WIRELESS SENSOR NETWORKS ..................................................................... 5
  2.2 TINYOS ............................................................................................................ 6
  2.3 DATA AGGREGATION ...................................................................................... 7
  2.4 DATA DECISION SYSTEM ............................................................................. 9
  2.5 HIDDEN MARKOV MODELS ......................................................................... 10
      2.5.1 Stochastic Process .................................................................................... 10
      2.5.2 HMM ...................................................................................................... 11
  2.6 LINUX SYSTEM INTERFACE SUBSYSTEMS ............................................... 13
  2.7 REQUIREMENTS SPECIFICATIONS .................................................................. 14
      2.7.1 Requirements of Simple Universal Gateway ........................................... 14
      2.7.2 Requirements of Smart Universal Gateway ............................................ 14
  2.8 RELATED WORK .............................................................................................. 15

3 Design and Implementation ....................................................................................... 16
  3.1 RESEARCH AND DEVELOPMENT METHODOLOGY ..................................... 16
  3.2 SOFTWARE DEVELOPMENT METHOD ........................................................... 18
  3.3 ARM LINUX SYSTEM DEVELOPMENT ......................................................... 19
  3.4 HEALTH CARE SYSTEM OVERVIEW ............................................................ 20
  3.5 HARDWARE AND SOFTWARE ARCHITECTURE ........................................... 21
      3.5.1 Base Station ............................................................................................ 22
      3.5.2 Smart Universal Gateway Development Board .................................... 23
      3.5.3 Boot Loader ............................................................................................ 25
      3.5.4 Root File System ..................................................................................... 26
      3.5.5 SQLite Database ...................................................................................... 26
      3.5.6 Work Flow model of Smart Gateway ..................................................... 27
      3.5.7 Packet Format between WSN and Gateway .......................................... 30
  3.6 SIMULATION OF SMART UNIVERSAL GATEWAY ........................................ 30
      3.6.1 Overview of IBM Rational Rhapsody as Simulation Tool ...................... 31
      3.6.2 Flow chart of Functional Smart Gateway .............................................. 31
      3.6.3 Hierarchical Use-Case Model ................................................................. 33
      3.6.4 Functional Use-Case Model .................................................................... 34
      3.6.5 File Diagram .......................................................................................... 35
      3.6.6 Message/Execution Diagram ................................................................. 41
  3.7 PROTOTYPING OF DDS ON EVALUATION BOARD .................................... 42
      3.7.1 Host PC setup for implementation ......................................................... 42
      3.7.2 File Management System on ARM Evaluation Board ......................... 44
      3.7.3 DDS Prototyping and Deployment on ARM Evaluation Board .............. 45

4 Results and Requirements Traceability .................................................................... 47

5 Conclusions and Discussions ................................................................................... 50

6 References ................................................................................................................ 51

7 Appendix .................................................................................................................. 53
  7.1 NFS setup on Host PC ..................................................................................... 54
  7.2 WSN Data Acquisition into ARM Evaluation Board ...................................... 55
  7.3 SQLite3 DATABASE COMPILATION & PORTING ON ARMBOARD ............. 56

VI
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Wireless Sensor Networks Block Diagram</td>
</tr>
<tr>
<td>2-2</td>
<td>Different Pointless and Promising Data Aggregation Cases</td>
</tr>
<tr>
<td>2-3</td>
<td>Working Diagram of Nodes and Base Station</td>
</tr>
<tr>
<td>2-4</td>
<td>Block Diagram of Decision System</td>
</tr>
<tr>
<td>2-5</td>
<td>HMM: Shaded nodes denote emitting state with output probability $b_i(v_j)$. The arcs represent state transitions with probability $a_{ij}$.</td>
</tr>
<tr>
<td>2-6</td>
<td>Concrete Example of Hidden Markov Model</td>
</tr>
<tr>
<td>2-7</td>
<td>Linux System on ARM Evaluation Board</td>
</tr>
<tr>
<td>2-8</td>
<td>Research Method and System Development Process</td>
</tr>
<tr>
<td>2-9</td>
<td>Water Flow Model for Software Development</td>
</tr>
<tr>
<td>2-10</td>
<td>Embedded System Development</td>
</tr>
<tr>
<td>2-11</td>
<td>Overview of Health Care System</td>
</tr>
<tr>
<td>2-12</td>
<td>Hardware Architecture of ARM Board</td>
</tr>
<tr>
<td>2-13</td>
<td>MIB520 USB Interface Board and (b) Interface Board Attached to Sink</td>
</tr>
<tr>
<td>2-14</td>
<td>ARM Evaluation Board Connection to Host PC</td>
</tr>
<tr>
<td>2-15</td>
<td>SQLite C/C++ API Flow Diagram</td>
</tr>
<tr>
<td>2-16</td>
<td>Flow Model With Simple Functions of Smart Gateway</td>
</tr>
<tr>
<td>2-17</td>
<td>Work Flow Model with Advanced Functions of Smart Gateway</td>
</tr>
<tr>
<td>2-18</td>
<td>Functional Flowchart of Smart Gateway</td>
</tr>
<tr>
<td>2-19</td>
<td>Hierarchical Use-Case Model</td>
</tr>
<tr>
<td>2-20</td>
<td>Functional Use Case Model</td>
</tr>
<tr>
<td>2-21</td>
<td>Abstract Universal Gateway File Diagram</td>
</tr>
<tr>
<td>2-22</td>
<td>Database_Util Functional Flowchart</td>
</tr>
<tr>
<td>2-23</td>
<td>Decision System Examples for Blood Pressure</td>
</tr>
<tr>
<td>2-24</td>
<td>WSN Timer Flowchart</td>
</tr>
<tr>
<td>2-25</td>
<td>ShowWSNData Functional Flow Chart</td>
</tr>
<tr>
<td>2-26</td>
<td>ShowWSNData Flowchart</td>
</tr>
<tr>
<td>2-27</td>
<td>SQLite Database Flowchart</td>
</tr>
<tr>
<td>2-28</td>
<td>Abstract Smart Gateway Simulation Results 1</td>
</tr>
<tr>
<td>2-29</td>
<td>Smart Gateway Simulation Results 2</td>
</tr>
<tr>
<td>2-30</td>
<td>ARM Board File System</td>
</tr>
<tr>
<td>2-31</td>
<td>Data Decision System Flow Model</td>
</tr>
<tr>
<td>2-32</td>
<td>SQLite3 Database Startup on ARM Board</td>
</tr>
<tr>
<td>2-33</td>
<td>ShowWSNData Flowchart</td>
</tr>
</tbody>
</table>

VII
1 Introduction

Thesis work comprises development of Data Decision System (DDS) prototype for the Universal Gateway, for e-Care health system. Data Decision System built in Universal Gateway does the functions such as collection of data from the elderly under surveillance in the form of Payload with the help of wireless sensor networks, aggregates and creates variable dependent alerts to the Health Care Centre.

Work is divided into two phases. First, DDS for the Universal Gateway is designed by considering Suitable activity recognition algorithms. DDS explains the state transitions, which is designed with respect to data from sensor nodes. Later data from sensor nodes determines how the alerts/alarms are acknowledged accordingly with respective to data driven from the wireless sensor networks.

Secondly, Prototype DDS Software Algorithm of Universal Gateway is designed, implemented and Prototyped on ARM-Linux platform.

Factors such as reliability, scalability of DDS in Universal Gateway for the wireless sensor networks are considered as Software design and engineering concepts during the system development process.

The aim of this thesis is to design and prototype the Data Decision System in the “Arm Evaluation Board”; a Smart Universal Gateway for Health Care System.

1.1 Background

Most of the modern Universal Gateways with Smart Data Decision System are compact, with sophisticated functionalities, fast performing and application specific.

The activity recognition data attributes such as temperature, humidity, pulse-rate, and heart-beat and so on, are retrieved with the help of wireless sensor networks deployed in specific application zones such as living beings, machines and other energy sources. This data from WSN motes is retrieved and formulated by using standard reliable De-aggregation and data decision techniques.

Activity Recognition Algorithms such as content based forwarding and routing are standard and reliable decision system design concepts. These are used in design and development of Data Decision System for health care sector, which makes Software Design and Implementation for the Embedded Intelligent Systems as always critical.
Implementation and Testing of DDS algorithm is performed on Samsung ARM evaluation board. Embedded Software Design, Simulation and Implementation, by the means of developing a HMM based Data Decision System for the Health Care System are the main concepts of thesis work ethic.

Thesis work carried out at WSN Application Labs, Jönköping Institute of Engineering, wills to contribute to e-care health monitoring systems.

Health Care Systems receives the data from the Motes installed with Elderly at home through different modes of transmission such as LAN, WAN or/and GSM/GPRS. Thus raises the necessity of the Smart Universal Gateway mediating the Elderly at home and E-care health centers by reporting the regular activities or health status of remote Elderly.

**Requirements Functions of Smart Gateway:**

Requirements functions for the prototype of the Smart Universal Gateway are explained in the further implementation section.

### 1.2 Purpose and aims

This thesis is a part of project “A Smart Gateway Design for WSN Health Care System”. The purpose of this thesis is to design software, build and test the data decision system for Universal Gateway. According to this idea, several questions are to be answered. Such as;

1. Assumptions during Software and Algorithm design?
3. Highlighting characteristics and Issues of Universal Gateway for Health Care?
4. Design and Implementation of Data Decision System?

#### 1.2.1 Requirements Functions of Smart Gateway:

- **UR-1** Acts as bridge between WSN and other various communication networks such as LAN, WAN.
- **UR-2** Receives sensor data from WSN, updating and storing in the database.
- **UR-3** Should consist of DDS to detect the real-time health states (i.e. normal, questionable, dangerous or oncoming dangerous) of the elderly depending on the current received data and historical data saved in onboard database.
UR-4 When suspicious activity from WSN of Elderly is detected, Data Decision System must enable gateway by sending notifications to the remote server. Also must send the emergency message to all the care-givers.

UR-5 Data Decision System must comprise of inbuilt server, saving the various health statistics of Elderly (blood pressure data acquisition in our case).

UR-6 Must report various statistics of the Elderly to the remote server and/or Care givers periodically.

1.2.2 Smart Universal Gateway Onboard system requirements:

UR-1 Smart Gateway server should comprise the Data Decision System and reliable database engine.

UR-2 Data Decision System should display real-time activity alarms based on the WSN data from Elderly under surveillance.

1.3 Delimits

Since the WSN Universal Gateway is in early stage of research and development, system development is simple with prototype definitions.

Data Decision System is developed based on assumption that only one elderly is at that home under surveillance, motes driving only single variable data i.e. the blood pressure.

Alarms, warnings and questioning event messages are not tested with either of LAN/WAN or GSM/GPRS.

Intelligent System Development security aspects are limitations, less focused on the quality of implementation and size, price and level of integration is less considered.

1.4 Outline

The below table illustrates the thesis outline;

<table>
<thead>
<tr>
<th>Chapter No</th>
<th>Key Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Theoretical background, Wireless sensor networks, TinyOS, Data Aggregation, Data Decision System, Hidden Markov Models, Linux</td>
</tr>
</tbody>
</table>
The rest of the article is structured as follows;

Following the introduction, chapter 2 illustrates theoretical background and problem formulation of the thesis work. The discussion about the data decision systems and the Hidden Markov Model algorithms is done. The specification Requirements of the Smart Universal Gateway are designed within this section.

Chapter-3 comprises of illustrates the possible research method adopted for carrying out the thesis work. Both system and software Engineering methods are highlighted during this section. Followed by the hardware architecture and software modelling and simulations using tools is done. Also consists of the implementation and prototyping of the Data Decision System on ARM Evaluation Board.

Chapter-4 comprises of the results, discussing and evaluating the simulation results and the prototyped Data Decision System.

Chapter-5 summarizes the thesis work carried out and points out related future work.

## 2 Theoretical background

The following theoretical meta-data illustrates the considered information used for conduction of the thesis work.
2.1 Wireless Sensor Networks

Wireless sensor networks are responsible for the modern smart environments [1], sensing data from multiple sensors of wide variety of modalities performing various functions. The challenges are enormous in the functions such as relevant quantity detection, data collection and monitoring, assessing and evaluating information, manipulating meaningful user display, decision making and alarm functions.

![Wireless Sensor Networks Block Diagram](image)

Figure 2-1 Wireless Sensor Networks Block Diagram

The above block diagram illustrates the conceptual framework of the wireless sensor networks and its application areas.

Whole WSN configuration is divided into two categories such as Data Acquisition System and Data Distribution System. The one of the main challenges of intelligent systems is to bridge the gap between these important zones. Thus raises the importance of the Smart Universal Gateway for WSN to meet the following challenges [2].

a. **Real Time Transmission:**
The Smart Universal Gateway must send from WSN networks to the destination address with no delay of time. This characteristic is considered very important in the health and defence sectors. Delay in transmission of intelligence from the WSN zones can lead to catastrophes such as machine failures or loss of life.

b. **Data Validity:**

Smart Universal gateway must exhibit consistent working performance. The system including hardware, software, operators and procedures, must check for correctness, meaningfulness and security of data that are input to intelligent systems.

c. **System Reliability:**

Discipline ensuring whether system is performing its intended functions in specific time length operated in specified manner. E.g., Built-in Data Decision System in Universal Gateway has to perform the functions by sending alarms and surveillance information of elderly to the caretakers or healthcare centres in real time and to remote places.

d. **System Scalability:**

A desirable characteristic of the intelligent system ensuring its ability to be readily enlarged or withstand the growing amounts of work. Data Decision System consists of functional programs and Database engine, must be compatible for the future work extensions either at software and hardware levels.

### 2.2 TinyOS

TinyOS is open-source component-based operating system and platform targeting wireless embedded sensor networks. TinyOS has a single stack thus making longer operation beyond few hundred micro-seconds have a callback. Its component based architecture enables fast implementation considering memory constraints inherent in sensor networks. TinyOS component library comprises network protocols, sensor drivers, data acquisition tools and distributed services – which can be built and reused for further custom application. TinyOS uses NesC; an optimized C programming language features to link the callbacks, called events, statically. TinyOS enhances programmers to write complex logics by merging many small event-handlers to maintain high concurrency.
During the design and development of Data Decision System in Smart Universal Gateway for Health care System prototype, Sensor node used all have installed with TinyOS version 2.0. It uses a standard message structure `message_t` which is a standard tinyos-2.x message buffer. Sensor nodes use this `message_t` frame structure to form both network route and send/receive messages. Frame structure of `message_t` is discussed in later chapter.

### 2.3 Data Aggregation

Data aggregation in [3] illustrated as the *systematic process of aggregating data from multiple sensors within the WSN zone to eliminate redundant transmission and provide fused information to the Base station.*

This is elaborated as the process of collecting the most relevant data from different sensor nodes of WSN, framing in a systematic order, fused at intermediate node and transmitted to the base station. The below diagram illustrates the different possible aggregation cases in WSN topology.

![Data Aggregation Diagram](image)

*Figure 2-2 Different Pointless and Promising Data Aggregation cases*
The WSN aggregated data packet format illustrated in later chapters explains the standard aggregation structure used for implementing the prototype model. Aggregation process characterises energy efficient transmission with minimum data latency.

![Working Diagrams of Nodes and Base Station](image)

**Figure 2-3** Working Diagrams of Nodes and Base Station

The diagram above shows the base station syncs the reliable node data in a systematic order and transmits to the recipients. The efficacy of the data aggregation can be measured by taking the following metrics into consideration.

**Accuracy** the most important metric – the difference between resulting values at the node to the true value. Conditionally, not all data is sent to the node any longer; the accuracy is expressed as ratios, differences, statistics and other values depending on particular case.

**Latency** aggregation can also increase latency of data reporting as intermediate nodes have to wait for data.

**Completeness** operational approximation obtained potentially is called completeness, the included percentage of all readings included in the aggregated final at the sink.

**Message overhead**, the main advantage of aggregation takes over in reduced overhead, results in the improvement of energy efficiency and network lifetime.
2.4 Data Decision System

In this section, DDS illustration is narrowed down to Clinical Health Care Support system assisting health care centres and caretakers of elderly at home. Decision support systems are computer application programs with decision making tasks, designed to assist physicians and other health professionals.

The below shown schematic diagram illustrates the sub-system blocks of data decision system [4].

![Block Diagram of Decision System](image)

**Knowledge Base** is a database for knowledge management, providing the means for collection of WSN data, organization and storage functions. Acquisition of WSN data in the form of payloads and deduce into logically consistent set of rules is its main operation. The logical reasoning methods such as Induction, Deduction, formulating the concerned results related to their problems and solutions.
**Inference Engine** does the functions such as drawing the conclusions by applying WSN data and information. Its main function is to extend knowledge base. WSN Inference is *systematic process of making conclusions out of WSN data which are subjected to random variation*. E.g. WSN payload constitutes the random variations due to low Quality of Service (QoS) and other observational errors from sensors.

**Communication Mechanism** comprises of hardware and software modules interfacing the decision system intelligence from the Embedded Device to the destined recipients such as healthcare centres and caregivers to elderly. Embedded System Device consists of upper-layer network protocol modules such as TCP/IP, CAN, HTTP, FTP and HTTPS.

### 2.5 Hidden Markov Models

Hidden Markov Models (HMM) are considered to one of the suitable clinical-activity recognition algorithm used by Wireless Sensor Networks. Elaborate description given from the context aware sensing concepts [5] adopting the Hidden Markov Model algorithms for the Activity Recognition of Elderly at home for E-Health Care Systems.

A Hidden Markov Model is a statistical Model, a simple Bayesian network built on sequence of variables, in which the system being modelled by assuming that the occurrence of unobserved states are with respect to Markov processes.

The Hidden Markov Model terminology explains the contexts of transition of hidden states of the simple Markov Model. The probabilistic transition of the state is not directly visible in this as of Markov Model but only output dependent on state is visible.

One of the context-aware sensing architectures is the HMM is described in the following section.

#### 2.5.1 Stochastic Process

According to [6], description of HMM is given by illustration of stochastic process and basics of Markov chain.

**Definition 2.5** A stochastic process is \( \{ X(t), \ t \in T \} \) is a family of random variable indexed by the parameter \( t \) in \( T \).

The above definition is the mathematical process illustrating the random variable involved over a parameter \( t \), where \( t \) is the Time and the set ‘\( T \)’ is referred as parameter space set.
Time factor with respect to the WSN zone data transfer characteristic can wither be discrete, this means sample data from the sensors are at equally spaced intervals or continuous. The sample space $S$ of $X(t)$ can be referred to as state space and elements of the states $Q$. Considering the $S$ space can be discrete in this case the total numbers of elements of $Q$ is either finite, countable infinite, or continuous considering the number of elements of $Q$ is uncountable infinitely many. The states are vectors or variables which the stochastic process can be in at any time.

So, there exist four different process models:

- A stochastic process with discrete state space and discrete time $t$.
- A stochastic process with discrete state space and continuous time $t$.
- A stochastic process with continuous state space and discrete time $t$.
- A stochastic process with continuous state space and discrete time $t$.

### 2.5.2 HMM

**Definition of** Hidden Markov Model of a stochastic system is a Markov chain [5] of the system plus some statistical model based on some observed data from a system.

According to the above definition, Hidden Markov Model chain contains of two parts. The first part is the Markov chain which is defined by $\theta = \{Q, A, \pi\}$, where $Q$ states of the Markov chain are not observable, thus hidden. And the second part is the $M$ set of observational symbols $Y = \{y_1, y_2, y_3, ..., y_M\}$ which are corresponded to the system output being modelled. This correspondence is represented by a matrix $B$, contains the probabilities of the emitting observations $y \in Y$ while in state $q \in Q$.

![Diagram of HMM](image)

**Figure 2-5** HMM: Shaded nodes denote emitting state with output probability $b_i(v_i)$, the arcs represent state transitions with probability $a_{ij}$
Hidden Markov Models implements complex Markov chains modelled on observational emitted states according to some probability distribution. The below concrete example illustrates the probabilistic state transitions limited to the possible health states of a human being.

Considering the present state of human being stated into set of two variables

\[
\text{State} = \{\text{Normal, Sick}\}
\]

The above two variables are considered to be probabilistic Markov states in the invisible to caretakers and further production states in Hidden State, invisible.

![Concrete Example of Hidden Markov Model](image)

**Figure 2-6  Concrete Example of Hidden Markov Model**

An elderly, who is under regular health surveillance by the health care system, does not have exact information about his health state, is subjected with the probabilistic occurrence states. Health care centre, considering the physical attributes such as age and other functional variables such as what job does he do and so on.

Initially, the care takers assume the transition of elderly from being NORMAL to SICK and vice-versa as {0.3, 0.4} probability respectively. Thus invisible transitions illustrate the functional attributes probably occurring under each set of states. Health care system assumes that Elderly under surveillance does only three activities such as going to hospital, office and gym. The choices of functions to do are determined by the health state of elderly.
The transition probability explains the change in health aspects of elderly as in the above shown Markov chain. The above Markov chain explains the possible functional activity that is performed by the elderly on each day.

Regular Activities = \{office, hospital, gym\}

On each day, the probable activities are performed by elderly depending on his health state. Elderly when NORMAL has 60% chance of going to hospital and when SICK has 10% chance of going to hospital.

2.6 Linux System Interface Subsystems

System interface is the collection of standards with interfaces and communication protocols with large number of devices. From Literature [7], Linux is an example of layered architecture that joins high-level drivers, such as disk-drivers, to physical interface such as fibre channel or serial attached system interfaces.
In the block diagram, the top level of kernel is the system call interface, which handles the routing of user space calls to their appropriate destination in the kernel (such as open, read, or write). The virtual file system (VFS), abstraction layer acts as the multitude of file systems supported in the kernel. VFS takes control of the routing request to the appropriate file system. The buffer cache optimizes the access to physical devices by caching recently used data. Device block layer consists of the various block drivers for the underlying devices.

At the low level, the device block layer is interfaced with the various low-level physical devices such as fibre channel devices; serial access system interfaces (SAS) devices, bridge and other communicating devices.

### 2.7 Requirements Specifications

The Universal Gateway runs in two modes:
1. Simple Mode with Database maintenance.
2. Smart Mode with Data Decision System (DDS).

#### 2.7.1 Requirements of Simple Universal Gateway

The requirements specifications of the simple WSN Gateway, when it is configured to the simple mode of operation is explained as in the below.

- Gateway has to send the **WSN Configuration** data and **GW Configuration** data from health care centre to the Wireless Sensor Networks at elderly.
- Upon the reception of the **Data Request** from the HCC, the Gateway must send back **Reception Ack** followed by WSN data to the Health Care Centre
- Upon the reception of **Data Stop** request from the HCC, the Gateway must stop receiving the WSN data from the Elderly and send **Ack** back to the health care system.

#### 2.7.2 Requirements of Smart Universal Gateway

- The Data Decision System must comprise a configured Database system within the Gateway.
- Universal Gateway must drive the **payload** data from nodes and save the de-aggregated data into the database file system.
- The payload from the WSN motes must be collected, updated and saved in the database, creating respective tables and files.
- Universal Gateway has to display the configuration of the Gateway in the terminal.
• If the Base Station between WSN and Universal gateway is disconnected, DSS should create an Alarm message “WSN Base Station is Inactive”.

• Data Decision System must receive the data from WSN network, must analyze, interpret data type and create the ALARMS and WARNING messages to the Care Givers of Elderly. The ALARM messages must be prioritized with respect to the data received from the sensor nodes; whether it is High Priority: Urgent Message, Medium Priority: Warning Message, Low Priority: Questionable events.

2.8 Related Work

The smart universal Gateway for the WSN health care system for the elderly at home, an interconnection and service management platform for home environments [8]. The work comprises of interfacing the smart universal gateway and health care centres through the Ethernet, Wi-Fi, GSM/GPRS communication modules integrated into smart Universal Gateway reporting information of Elderly to the Care-givers. The result interprets the reliable, low latency and low power consumption characteristics into highlight.

The cases [9] [10] such as gateway placement, disaster management, combat field reconnaissance and secure installations are important due to large population sensors, working on small batteries. In the paper [11], genetic algorithm for hop count optimization, also distance optimization method for best spot selection for WSN Gateway for each sensor node group so that the data can be transferred to gateway with minimum latency. For our thesis, Gateway design for elderly people living in his home has low latency because of short distances within sensor nodes placement.

Communication between the different WSNs with diverse protocol for the routing, data acquisition and other applications must communicate with the multi-TCP/IP protocol such as IPV4 and IPv6. Thus raises the need of compatible modern WSN network protocol which is based on application-level gateway concept. According to authors [12], the services and protocols for both WSN and Internet are most challenging tasks for gateway design, providing services and protocols. The gateway design should connect heterogeneous networks by connecting different networks, providing protocol and service translations to different combination of protocols on both sides of Gateway. Network protocol building, configuration and online updating can be very good features for a gateway design to increase Gateway compatibility with the modern WSN protocols.
Authors in [13] illustrated the proposal regarding the online sensor data access service in Gateway to the users with great convenience. A light weight network nodes interacting approach; based on Web 2.0, they tried to maximize reuse the shared devices using standard Web technologies. The design and implementation of web-based interaction and management in gateway can be one of the future works providing remote data access to WSN.

3 Design and Implementation

In this chapter, we elaborately discuss about the research and development methodology concepts for the development of thesis.

3.1 Research and Development Methodology

The research methodology adopted for the development of the thesis work is system development methodology [14]. The system development methodology comprises of 3 major steps: concept development, system building and system evaluation. The thesis begins with the concept of developing a Data Decision System in Smart Universal Gateway.

The consecutive phase at system evaluation includes the design, implementation with the exploration of the specification requirements of the system. Once the system functionalities are defined, they are evaluated through Design, simulations and implementation of developed functionalities of the system. Theory testing is the main aspect of system development method, used properly to implement and test the DDS in Universal gateway where System development method is stacked with iterative steps, combined with actual situation.

Data Decision System for the Smart Universal Gateway is illustrated as below.
Thesis work started with an idea of scientific, philosophical belief that there is a requirement of an intelligent system which is smart, making its own decisions out of the acquired data from the WSN nodes located at the elderly at home and facilitate with remote service data provision service to the HCC or care giver.

During the concept building stage, the various questions regarding the development of DDS such as what are suitable activity recognition algorithms, what are the specifications requirements of the system that has to build? The requirements specifications for the development of the DDS for the Universal Gateway are taken, simulated and analyzed.

The later phase consists of the DSS implementation and prototype on intelligent system and validates the results with respect to the simulation results at the end of the simulation phase. *In the thesis work, the emphasis of choosing system development illustrates the concept of system development and theory illustration by the DDS in Smart Universal Gateway within the field of Wireless Sensor networks but on the quality on the system implementation.*
3.2 Software Development Method

The software development method used for the building the Data Decision System based on the Hidden Markov Model is of Water Flow Model. Water flow model is a sequential software development process, the flowing steadily downwards like a waterfall.

![Water Flow Model for Software Development](image)

Figure 3-2  Water Flow Model for Software Development

The above is the Hardware-Oriented methodology for the adapted for the software development. To follow water-flow model, one proceeds from finishing one step completely from top to the bottom and the approach is irreversible. Once the software requirements phase is completely finished, next phase is drawn in. This model is adapted when the projects are stable, likely that designers can predict the problem areas completely. The implementation process is carried only with complete system design awareness and complete design.

The System development technique used is the **Prototyping**. Where only the partial functionalities of the system are implemented and evaluated. The purpose of the prototype is to allow users to get acquainted with the idea of design rather than implement, test and evaluate the actual system based on description.
3.3 ARM Linux System Development

This section explains the embedded system based application development process. Firstly, the development starts in a host computer with the development tools relating to Embedded Devices which are under implementation. After the use of simulation tools and evaluation board for the debugging, the generated image file on the host computer is integrated into the target embedded device.

![Diagram of Embedded System Development](image)

Figure 3-3  Embedded System Development

Above diagram illustrates the Embedded System development listing a few important characteristics.

- Embedded Intelligent Systems are built for specific application purpose performing multiple tasks considering real-time constraints and operational issues of usability and reliability of the embedded device.
- Embedded Intelligent systems are prone to hardware limitations such as little memory, and keyboard interfaces.
- The runnable files or program instructions are saved in the flash memory, firmware.

General 3 principal steps involved during the ARM embedded system development are listed as following.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Deriving objectives of hardware such as Flash memory, communication ports, peripherals selection etc.</td>
</tr>
<tr>
<td>2</td>
<td>System Development Requirements such as program development, kernel porting, file system development and hardware drivers.</td>
</tr>
<tr>
<td>3</td>
<td>Developing applications.</td>
</tr>
</tbody>
</table>
3.4 Health Care System Overview

The below picture illustrates the application setup of smart gateway for the elderly at home to the Care Givers, clinical service providers who receives the elderly Meta data from remote locations.

![Overview of Health Care System](image)

Figure 3-4  Overview of Health Care System

The thesis work is mainly focused on prototyping the smart Gateway, which receives the WSN data from the nodes located with the elderly at remote location. As mentioned early, the Design, simulate and implement to prototype the specification requirements of Smart Gateway is the main task of this thesis.

Whole architecture of the system consists of 3 sections. The first section is the WSN at the Elderly at home. The WSN nodes collect data from various attributes of Elderly, aggregated at the Base-station and sent into the Smart gateway. Second section comprises of Smart Universal Gateway which consists of 2 sub-systems such as Data Decision System and Database Engine. Data Decision System will manipulate the data driven from elderly WSN and Database engine is used to create data files, save and update the WSN data. DDS also does the functions such as generating and sending alarms to the health care centres through the public communication networks including internet, GSM/GPRS, Ethernet and Wi-Fi.
3.5 Hardware and Software Architecture

This section illustrates about the ARM hardware evaluation board used for prototyping the Data Decision System. The below block diagram explains about the three sub-units such as Central Unit and WSN connectivity to the ARM evaluation board and to remote care takers through various communication interface.

![Diagram of Hardware Architecture of ARM Board]

As there are many WSN modules available to connect sink node or base –station to the Smart Universal Gateway to prototype, we choose to use the MICAz MPR2400 manufactured by Crossbow Company. Since this thesis work is the part of Smart Gateway Design proposal from the WSN Health Care Research Group of Jönköping University, considering compatibility constraints with other sub modules of Health Care System. Crossbow provides interface boards such as MIB 510, MIB 520 and MIB 600 interface boards with different communication connectivity interfaces. MIB510 uses serial interfacing, MIB520 uses USB interface board and MIB 600 uses Ethernet Interfacing Board. Comparing MIB520 and MIB 510, to match the present WSN module prototype, we chose MIB520 because it has better performance such as high data transfer rate, plug and play, small size and large number of ports. MIB 600 must use DHCP function, which will take more memory space, making the system structure more complicate and high power consumption. By considering above comprehensives, we decided to use MIB520 in the WSN Module.
3.5.1 Base Station

In our Smart Gateway design, we use MIB 520 USB interface board acting as Base-Station between the WSN and Smart Gateway with 56.7K baud rate. MIB520 has USB connectivity support to MICA motes for the communication and programming.

FTDI FT2232C chip used by the MIB520 board allows Gateway to use USB as Virtual COM port. Host device requires FT2232 chip driver to communicate with MIB520, where host device can read and write the USB bus in serial Mode.

Considering the port sequence; First port is dedicated to in system mote programming, where MIB520 has an On-Board in-system processor (ISP) – connected through MICA series connector. An Atmega16L situated at U14 used to program the motes. Functionally, the ISP programs the code into sensor node through USB port to the host computer with the installed Mote works and TinyOS. This development and debugging method has too heavy load on Smart Gateway Board, which is handled easily by host PC.

Second port is used for Data Communication with host device over USB is achieved by connecting MIB520 to WSN by attaching the sensor node to its MICA-series connector. The above figure explains the combination of MIB520 and sensor node forming a base-station, which will detect all messages having the destination address of host device and sent it over MICA-Series connector to MIB520 through USB bus to host PC device.
The above picture illustrates the ARM Evaluation Board with the Data Decision System Connecting the Base station and Host PC via debugging connections.

3.5.2 Smart Universal Gateway Development Board

The below shown is the S3C2410 development board to implement the centre control unit for the smart universal Gateway.
This board use SAMSUNG S3C2410 core microprocessor, providing general applications and hand held devices in the cost effective, low-power and high-performance micro-controller solutions in a small die size.

S3C2410 micro-processor has many vital features and high price-to-performance ratio. It is developed using ARM920T core, using 0.18um CMOS cell and a memory. The ARM920T implements MMU, AMBA BUS, 5 steps pipeline, with distinguished 16KB instruction and 16KB data caches, each with an 8-word line length.

The functional support features of the hardware are that it provides 200 MHz (266 MHz) standard operating frequency and 64-way set-associative cache with I-cache (16KB) and D-Cache (16KB). S3C2410 core board used provides 64MByte SDRAM and 64MByte NAND Flash onboard, 32bits bus width and 100MHz front side bus. This firmware guarantees all application programs can be saved onboard and executed with fast speed.

The development board comprises of Ethernet controller (CS8900A) facilitating a RJ45 10BASE-T Ethernet interface, helping the smart gateway to connect to local PC or LAN; Two RS232 Serial UART ports for GSM/GPRS terminal connection and computer console; Two USB host for MIB520 module and external memory to DB; JTAG debugging / programming interface.
3.5.3 Boot Loader

Bootloader is the first paragraph of the code that runs after the system power up. It is equivalent to the BIOS of PC. To put it simply, bootloader first load the Linux kernel from memory to RAM. Then it initializes all the necessary hardware devices on board. During this step, some messages which are required by the system kernel are created and passed to kernel through relevant mechanisms. This will bring the system hardware and software environment to a proper state. Last thing it does is system test and give the control of the development board to Linux operation system.

In our S3C2410 development board, we use the open-source bootloader program VIVI, which is developed by MIZI Company in South Korean. Base functions of this VIVI are as follows:

- Download image file, like OS kernel and root file system, into memory through Serial port or network.
- Setup the system boot up parameters.
- Initial hardware and boot up the operation system.
- Memory partition and bad block detect.
- Boot up delay settings.

When doing memory partition, most of attention should be paid to the application space, must be high. In few embedded system, root file system and user application are segmented into different partitions. This process will waste some space of the memory, which is considered as drawback for prototyping the Gateway Design, because of memory space wastage. Also, size of other partitions must not be set bigger than its original size. After measuring the size of VIVI, system boot up parameters and Linux kernel, partition table for our smart gateway design are listed as below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Offset</th>
<th>SIZE</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIVI</td>
<td>0x00000000</td>
<td>0x00020000 (128 KByte)</td>
<td>0</td>
</tr>
<tr>
<td>Param</td>
<td>0x00020000</td>
<td>0x00010000 (64 KByte)</td>
<td>0</td>
</tr>
<tr>
<td>Kernel</td>
<td>0x00040000</td>
<td>0x001c0000 (1.768 MByte)</td>
<td>0</td>
</tr>
<tr>
<td>Root</td>
<td>0x00200000</td>
<td>0x03cf8000 (60.992 MByte)</td>
<td>16</td>
</tr>
</tbody>
</table>

Figure 3-3 Memory Table Partition
3.5.4 Root File System

Root File System is an important component for Linux system boot up, and also necessary for normal operation of the operating system. Kernel code and image files are stored in root filesystem. When system boot up, kernel will load the root file system to RAM from NAND flash, and then mount the necessary system devices into it. The original Root File system installed in development board is a Compressed RAM File System (CRAMFS) which follows Filesystem Hierarchy Standard (FHS) and was made by BusyBox 1.0. Standard folders such as `bin`, `dev`, `etc`, `lib`, `usr`; and common commands like `ls`, `vi`, `cat`, `mount`, `tar` etc are included in this filesystem. The shortcoming of the filesystem is that the C language library which is of version 2.2.2 is updated to 2.3.2 by updating link files in folder `lib` in order to meet the requirements of setting of DDS, porting and running of SQLite Database engine. We use NFS (Network Files System) to test our program.

3.5.5 SQLite Database

Database engine ported on the ARM evaluation board is SQLite version 3. The reasons for opting SQLite as the Database engine are as following sophisticated facilities that it provides.

- **Zero Configurations**: Initial No-Configuration is required to setup the database. Single binary file with the included library files act as pre-configured database engine, with enough features.

- **Portability**: It is built with consideration of portability issues. Database can be complied on windows and other operating systems such as Linux, Solaris, Mac and other embedded platforms such as QNX, VxWorks, Symbion, and PalmOS etc. It works on 16-, 32- 64- bit architectures with both big-endian and little-endian byte orders. Database file binary format is compatible to all supported Operating Systems. Upon to OS limited capacity, it can hold up to 2 terabyte of data. And supports both UTF-8 and UTF-16 encoding.

- **Simplicity**: SQLite database is built simple; its API is easy to use. It is customized to build our own convenient functions in C. It has vast language libraries related to Python, Perl, Ruby and Java etc. Each module in it is specialized, performing a specific task. Its each module is developed independently and to debug queries as they pass from one module to other. It is also easy to add new features, debug and gain database reliability.
Flexibility:
SQLite is a flexible database. Being a light database, it provides both facilities such as power and flexibility of a relational database frontend. It requires no database servers to configure, no networking or connectivity problems, no platform limitations, no license fees.

3.5.6 Work Flow model of Smart Gateway
The following below is the work flow diagram of the Simple Gateway Model
The above work model explains the data flow functions of the Universal Gateway for the Wireless Sensor Networks. The most apparent functionalities are comprised with Data Decision System, which manipulates and draws the decisions such as warning messages and alarms out of the WSN data of elderly from nodes. The above model does not consist of the Database Management and Data Decision System in the Gateway.

The Gateway in the above simple model has to send the configuration details of WSN, GW to the Health care Centre on acquisition of **GW configuration req**, **WSN configuration req**. When the data from WSN is ready, the Gateway has to send the **Ack** to the HCC about the data reception and send WSN data to the health care. Health Care Centre sends back **Receive Ack** back to Gateway and Gateway will stop sending WSN data to the HCC when it receives the **Data Stop** command from HCC.

The following below is the hand drawn work flow diagram of the Smart Gateway Model.
Figure 3-11 Work Flow model with advanced Functions of Smart Gateway

The above work flow diagram illustrates the Simple Gateway Model with added functionalities such as the Data Decision System, Database Manager collectively called as Smart Universal Gateway. The Smart Universal Gateway must be able to send the **GW Configuration**, **WSN Configurations** to the WSN from the Receiving Station and **ACK** back to Receiving Station.

With the initialization, the gateway has to receive **payload** from the WSN, de-aggregate the data format, create the database tables and save. The data from then must be manipulated accordingly with Data Decision System and create alarms with respect to the variable attributes of which sensor nodes are meant to drive information or data from the Elderly. Variables such as blood pressure, pulse-rate etc are the example attributes of elderly of which sensor nodes are deployed to collect and send it to the Smart Universal Gateway. DDS of the Smart universal Gateway must provide the functionalities such as sending **Alarms, Warnings and Critical** message information to the receiving station. DDS must also able to tell the receiving station if the connection is lost between the Smart Universal Gateway and WSN zone deployed with the Elderly.

When Smart Universal Gateway receives the **Normal Health Report** request from the Receiving station, the most phenomenal function to be performed is to send the health report to the receiving station.
3.5.7 Packet Format between WSN and Gateway

Sensor nodes used in this research have TinyOS-2.0 installed. In TinyOS-2.x, the standard message buffer is called message_t. Frame structure of message_t is shown as below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Destination Address</th>
<th>Link Source Address</th>
<th>Message Length</th>
<th>Group ID</th>
<th>Active Message Handler Type</th>
<th>Payload Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size (bytes)</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>User Design</td>
</tr>
</tbody>
</table>

Frame head (from destination address to active message type) is used for packet routing in WSN. They are useless to the HCC. Only the “Payload” segment is useful for smart gateway or HCC. We consider the “Payload” segment design is very important to the smart gateway design because, first, it realizes the functions of WSN healthcare system, and second, the data package has the largest transmits number in the WSN. Its designed significantly affects the overall system functionality and efficiency. We designed the Payload format is as shown in below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Mote ID</th>
<th>Counting</th>
<th>Data Type</th>
<th>Data Value</th>
<th>Parent ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

We try to keep this payload segment design as efficient as possible. “Mote ID” segment indicates the packet sender’s id. “Counting” segment is the packet sequence number or time stamp. “Data Type” identified the type of data packets. We have considered nine different kinds of data packet for both home sensor network (HSN) and body sensor network (BSN) in our health care system prototype including heart-beat rate, temperature, microphone, Received Signal Strength Indication (RSSI) etc. “Value” segment represents the data type value got by the sensor. “Parent ID” segment indicates the mote ID of the sensor node of the last hop in its transmission path.

3.6 Simulation of Smart Universal Gateway

During this section, we discuss about the modelling of Smart Universal Gateway, Design, Simulation and Software Tools used.
3.6.1 Overview of IBM Rational Rhapsody as Simulation Tool

Since many Embedded Software Engineers are using standard modelling language with aspects of Model-Driven development [15] process to improve quality, productivity and communication. This can be used to make easy design, ease error corrections and other cost related issues.

This tool has an integrated software development environment which helps us to create, test, document and deliver the applications of software for the embedded real-time applications.

Some of the work features of the simulation tool are as following:

- Improved support for safety critical software development by supporting the MISRA-C, 2004; modelling and analysis of real-time embedded system’s profile and by embedding requirements text directly in the code generated.
- Facilitation of a Code-Engineering with project or code base integrated, providing code generation updates to files, seamless code visualization, reverse-engineering and code-visualization to flowcharts.
- Enhanced SysML 1.1 support and improvements of activity diagrams with functional flow definitions of system.
- The options such as Code Visualization and Code Update, we can develop code according to visual model, updates the code accordingly to the continuous change reflections. The modelling and simulations are meant to generate C code, considering implementation nearness aspects.

3.6.2 Flow chart of Functional Smart Gateway

The below shown is the functional flow chart of the smart Gateway with the Data Decision System prototype, used to model a quasi-functional clinical decision system design for the E-care health system.

The initiation of WSN Smart Universal Gateway starts with WSN module, where MIB520 base-station starts receiving data from different sensor nodes of the remote WSN at elderly. This way, the database system gets initialized, creating data files and updating them with real time WSN data from base-station.
Figure 3-12  Functional Flowchart of Smart Gateway
Since SQLite database files system works as one can write into file and many can read from the file, parallel modules as executable binaries, run in parallel on evaluation board. DDS takes data from the file system in real time mode, manipulates and creates alarms. The destined receivers of e-Care health system are care givers and health care centres.

It also performs function such as to send sensor information bulk information to the health care system on request. In the above block diagram, the events generated by the health care centre requesting the WSN configuration, file acquisition requests are not mentioned above. These events generated from the external triggering taken into consideration while undergoing simulation process. In this section we discuss with the Data Decision System design Modelling and Simulation using the Rhapsody Tool. The below Diagram Illustrates the different functional blocks created as the sub-systems with the Smart Gateway for the E-care Health System as part of modelling system.

3.6.3 Hierarchical Use-Case Model

![Hierarchical Use-Case Diagram]

Figure 3-13 Hierarchical Use-Case Model

The above hierarchical model illustrates the components of the smart Universal Gateway and their inter-dependencies. Universal Gateway block implements other three sub-components such as DatabaseAdministration, WSNetwork and HealthCareCentre. Explanation of individual blocks is given in the functional model, which includes internal functional processes and tasks.
3.6.4 Functional Use-Case Model

The below functional use-case model illustrates the Use case model of smart Universal Gateway application for the healthcare system. The application diagram illustrates the chain of blocks;

- **WSN of Elderly at home** is the block explaining about the WSN motes deployed at elderly at home, collecting the information of physical attributes such as Displacement of Elderly, Pulse rate, heartbeat, room temperature, humidity etc. The collection of various physical attributes data is done by Data Acquisition System which is explained in the later sections.

- **Data Acquisition System** is the data aggregation system built by using the various data collecting motes and aggregated at the base station. During thesis work, we used pre-configured aggregation devices such as crossbow sensor motes and base station. These hardware data aggregation devices use TinyOS operating system and programmed using NesC language. NesC consists of in-built methods and functions for application development. The Data from the sensor nodes is collected and sent to the smart Universal Gateway.
• **Smart Universal Gateway** is the functional block consisting of 3 sub-major blocks: Data de-aggregation, Data decision system and Database management system. Data from motes is saved by database management system deployed in the ARM Evaluation Board and is manipulated further by the DDS block. The DDS system de-aggregates the data from the sensor node packet structure, pre-scales and creates the Alarms and Priority messages and displayed to Health Care Centre.

### 3.6.5 File Diagram

![Abstract Universal Gateway File Diagram](image)

Figure 3-15 Abstract Universal Gateway File Diagram

The above shown file diagram comprises the 3 sub-systems, acting as the functions in a smart gateway are

1. **UniversalGateway**
2. **HealthCareCentre**
3. **WSNNetwork**
4. **DatabaseAdministration**

The **UniversalGateway** system block holds the functions and each very specific action performed by the function are illustrated as below.

a. fnCollectPayload(): This function acts as the base-station driver, collecting WSN data from Elderly and Saving as the raw payload structure, creating files in the database.
b. Timer(): This function illustrates allows smart Universal Gateway to check for the USB connectivity, if connection is not successful, the smart Gateway has to reset itself with certain clock time delay.

c. fnDatabase_Util(): This function creates the new file with local machine time, updates and cleans the trash when the tables are removed or deleted.

d. FnDataDecisionSystem(): This function takes the data from database, de-aggregates, makes decisions and generates the alarms, warning messages respective to the Data.

e. evSysON(): This function demonstrates the ON and OFF tasks of the Smart Universal Gateway.

The HealthCareCentre block illustrates the following functions;

a. showWSNData(): The respective function displays the retrieved WSN data to the destiny or the concerned recipients such as caregivers or health care centres.

b. showUserProfile(): This function displays the user profile of elderly who is under surveillance of WSN network. In our case, we implemented considering that there is only one elder under WSN surveillance and retrieves the data of single attribute i.e. Blood Pressure.

c. sendAlarm(): This function sends/displays Alarm signals derived from the DDS.

d. WSNOFF(): This function intimates the recipient/ Health Care Centre when the DDS system stops retrieving data from the Base station of WSN network deployed at Elderly.

The WSNNetwork block illustrates the following mentioned functionalities;

a. aggregateData(): This function displays the aggregated payload from the WSN network deployed in our case. In our case, we used pre-built base-station and motes which aggregate data within the base station built on TinyOS platform and Programmed on NesC.

b. collectPayload(): This function collects payload from the base-station and stores in the database ported in the Smart Gateway.

c. evTrigger(): It is the function which manipulates the external event triggers such as profileRequest(), sendWSNdata() etc.

The DatabaseAdministration block comprises of database related functions such as openTable(), closed(), updateDB(), cleanJunk() and callBack().
Sub-Files of Universal Gateway File Diagram

As mentioned above, the file diagram of the Universal Gateway has four sub-files. In this section, we will know the process of sub-components used in each block of files.

3.6.5.1 Universal Gateway Files

The Universal Gateway consists of the three functional flowchart diagrams as shown below.

1. Database_Util Function Flowchart

![Figure 3-16 Database_Util Functional Flowchart](image)

The above Database_Util functional flowchart illustrates how the Ported Sqlite3 Database in the evaluation board is used for the DDS. The Payload from the base-station is of big-endian format, is transformed into default data type, stored in the database, updated and closed. Payload structure is described in the theoretical background section, which comprises of MotelID, DataType and DataValue.

2. DataDecisionSystem Functional Flowchart: The below diagram illustrates the Hidden Markov Model based Data Decision System (DDS) driving the single variable attribute, i.e. Human Blood Pressure. The hidden stages of the variable readings are derived
The blood pressure levels comprise of both systolic and diastolic stages and their respective probabilistic sub levels of Decision System. Each and every stage is assigned with Display Notification during prototyping.

3. WSN Timer Flowchart:
The below flow chart illustrates the timer triggers self when the base-station connection get failed. Periodically, this timer checks the WSN connectivity with the Smart Universal Gateway. If the connection gets failed, the signal display intimates the recipients about the No-Connectivity problem.

### 3.6.5.2 Health Care System Files

1. ShowWSNDATA Function

The above flow chart illustrates the process of data acquisition and storage of WSN payload data in SQLite3 database in the evaluation board. WSN data is displayed when **WSNDATAREQUEST** event is triggered. The payload conversions are done within this function.

2. ShowWSNDATA Function

---

Figure 3-19 ShowWSNDATA Functional flow chart
Figure 3-20 show WSN Data Flowchart

The above flowchart does 2 tasks such as retrieving the profile information of the elderly under WSN surveillance in the SQLite3 database. Updates, displays and cleans the junk under deletion. Another task, when it receives the ProfileReqEvent event externally, it will retrieve the userProfile Information saved in the database and displays or sends it to the Health Care Centre.

3.6.5.3 Database Administration State chart Files

a. SQLite3 Database State Chart

Figure 3-21 SQLite Database Flowchart
The above illustrates the flow model of the SQLite3 database ported in the ARM Evaluation Board. Database utilization comprises of the stages such as Database Initiation, OpenTable, callback, updateDatabase, cleanJunk and CloseDatabase.

### 3.6.5.4 WSN Networks

It is the network of motes deployed at elder’s home, retrieving the physical data of elder’s environment or his own such as the physical location, movement, humidity, body temperature, pulse rate and blood pressure etc.

### 3.6.6 Message/Execution Diagram

![Execution Diagram](image)

**Figure 3-22 Smart Gateway Simulation results 1**

The above execution diagram is an animation of simulation output. With the evSysOn(), the smart Universal Gateway System turns ON. The aggregated data from the Base-Station is driven into the Smart Universal Gateway. WSNOFF or WSNON alarms are displayed if the Base-Station connection to the Smart Universal Gateway is OFF or ON respectively.

1) When the Gateway System is ON, the Database_Utility retrieves the data from the WSN nodes, opens the tables created, updates it with the data from them and closes Database.
2) The Data Decision System manipulates the data saved in the database, creates Alarms and Messages, and will display it to the Care Centre.

![Diagram of Smart Gateway Simulation Results 2](image)

Figure 3-23 Smart Gateway Simulation Results 2

The above simulation execution diagram illustrates the message transitions, showing the activities performed on different events generated from HCC. Upon it, the gateway performs the tasks by sending alarms and messages from the WSN network at elderly to the caregivers.

### 3.7 Prototyping of DDS on Evaluation Board

The main sections involved in prototyping process and steps of implementation are illustrated under this section.

1. Host PC setup for implementation.
2. DDS application development and Deployment on ARM evaluation board.
3. Functional Testing of DDS on ARM evaluation Board.
3.7.1 Host PC setup for implementation

During this section, software flow process and component selection are taken into consideration. Host PC is setup to cross-compilation of the DDS applications which are to be deployed on the Arm-Linux board.

Factors considered for Host PC setup for building ARM Linux Application binaries rather than building directly on the targeted platform:

- The probable important factor is the speed, because Host PC is most of the time faster than embedded devices for compilation.
- Another big problem is that a complete compilation tool-chain does take a great amount of disk space. And this space is usually very limited on embedded systems.
- Sometimes, lacks enough memory space to compile big files in the embedded system.

3.7.1.1 Cross Compiler Setup

The step by step SQLite3 Database cross compilation and NFS (Network File System) setup on host pc is explained as shown in Appendix, considering the various pre-requisite various software utilities. The host PC operating system chosen is Ubuntu Linux; GNU/Linux Kernel 2.6.24-generic release. Literature [16] explains the how the Cross compiler toolchain are setup on the host PC

3.7.1.2 SQLite Database setup on Host PC

The description of SQLite database is given in the previous chapters. Here, we illustrate the SQLite database setup [17] on the host PC. The below illustrates the simple step by step process of cross-compilation and porting of SQLite3 Database on the Host PC.

1. The source code is downloaded from www.sqlite.org. Distribution used is sqlite-3.5.x.tar.gz, downloaded and placed it in the temporary directory (e.g, C:\Temp).
2. Navigate to the temporary directory using UNIX file system conventions.
3. Unpack the SQLite tarball, Issue this command:
tar –xzvf sqlite-3.5.2.tar.gz

4. Move to the un-packed directory
   cd sqlite-3.5.2

5. Create the Makefile, for multithreaded DLL, run
   ./configure --enable-threads

6. Build the source:
   make

7. Create the SQLite DLL:
   dllwrap --dllname sqlite3.dll --def sqlite3.def *.o

8. Create the import library:
   dlltool --def sqlite3.def --dllname sqlite3.dll --output-lib sqlite3.lib

The ARM cross-compilation is done considering linking the SQLite3 Database on HostPC. While cross compiling the DDS application for porting on the ARM Board, the Libraries and Include files of the SQLite3 Database are linked as in below. Later, the compiled executive binaries are ported into the ARM Evaluation Board.

3.7.2 File Management System on ARM Evaluation Board

Serial terminal setup for the file system management between Host PC and ARM evaluation board is explained in the section. The following illustrates the File Management System on ARM Board. Cable used to interact between ARM board and host PC is RS232-USB.

The Network File System of the ARM Evaluation board is utilized and Implemented by clone emulator of MS-DOS "Telix Communication Program" known as Minicom:version 2.3. It is the menu program which emulates ANSI and VT102 terminals.

Minicom is installed on Host PC from the Repositories available for the Ubuntu version Linux. Configuration of the ports to the ARM evaluation board is set default to USB device port (Port /dev/ttyUSB0).
ARM Evaluation Board File System appears as below shown;

```
Looking up port of RPC 100003/2 on 192.168.0.3
Looking up port of RPC 100005/1 on 192.168.0.3
VFS: Mounted root (nfs filesystem).
Mounted devfs on /dev
Freeing init memory: 104K
mount /etc as ramfs
re-create the /etc/mtab entries
init started: BusyBox v1.00-pre2 (2007.07.06-03:31+0000) multi-call binary
.: 22: Can’t open /etc/profile
Please press Enter to activate this console.
BusyBox v1.00-pre2 (2007.07.06-03:31+0000) Built-in shell (ash)
Enter ‘help’ for a list of built-in commands.

#
bin etc lib linuxrc proc symbol usr
dev home lib-old mnt sbin tmp var
#
```

Figure 3-24 ARM Board File System

The cross-compiled DDS and Sqlite3 database from the Host PC are ported in the `usr` directory and Tested.

### 3.7.3 DDS Prototyping and Deployment on ARM Evaluation Board

In this section, we will know about the data decision system application which will be built on HostPC and Deployed on the ARM Evaluation Board. The Data Decision System Application on the ARM Evaluation Board is designed keeping insight of Database; Sqlite3 for the implementation.
3.7.3.1 **SQLite Database porting on ARM evaluation board**

From the literature [18], the step by step cross compilation procedure is followed and ported on the ARM Linux Evaluation Board. The SQLite3 Database comprises of single WRITE and multiple READ.

The SQLite3 database is ported in the directory: `usr/Sqlite`. Cross-compiled DDS binaries are ported into the same directory and tested.

**3.7.3.2 Building DDS prototype and deployment on ARM target board**

Considering the figure-32, the DDS for the Universal Gateway is designed by using the several considerations such as Cross Compilation of the DDS applications on the Host PC, Porting and Testing on ARM evaluation board.

The data from the base-station is driven into Universal Gateway, saved and updated real-time in the database. The decision system application binary is another application which utilizes the WSN data saved in the SQLite3 Database ported in the ARM Evaluation Board.

The WSNAcquisition Binary present in the directory (usr/sqlite/) of the ARM Evaluation board performs the task of data acquisition from the WSN network nodes collecting the data from the elderly under surveillance.
Figure 3-27 WSN Payload Bit stream in ARM Board

The above is the payload data packet frame received from the WSN base station into the Universal Gateway. The Base Station, MIB520 drives the WSN data and saves payload information in the SQLite database.

During the thesis prototyping, administration of the SQLite database is not taken into to the wide consideration.

Decision System is another application binary which uses the payload data from the SQLite database and creates the alarms by displaying the messages. The testing of DDS is done by considering display of messages using the Hidden Markov Model Algorithm adopted for decision making.

4 Results and Requirements Traceability

In this section we discuss about the experiment output results and traceability of the requirements. The results obtained by running the built application binaries in the terminal of ARM evaluation board are shown. The functional specifications requirements designed to prototype are traced.

4.1 Display outputs on evaluation board

When the DDS is deployed in the Smart Universal Gateway i.e. the ARM Evaluation Board, the following are the functional results of the Data Decision System.
1. The WSNAquision application binary drives the payload information of the motes, collecting the physical data of the elderly under surveillance and saves it in the SQLite database.

2. Followed by, the DecisionSystem application binary takes the updated data from the SQLite Database and implements the Hidden Markov Algorithm

- Displays the user profile of the Elderly created for motes deployed, e.g. displaying the Name, Address of the Elderly upon the start of WSN data acquisition.

- Pre-scaling of the payload information is done, Deducing the Alarms and Priority information out of the data received from the elderly. The alarm sets are probabilistically chosen.

<table>
<thead>
<tr>
<th>Elderly’s Name: Dedor Behanet,</th>
<th>Age -60, E44</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ostratorp, 52345</td>
<td>Skövde, Sverige,</td>
</tr>
<tr>
<td>Data Acquisition : Blood Pressure, Pulse Rate</td>
<td></td>
</tr>
</tbody>
</table>

**High Priority**: SMS and Voicemail to Health Care

**Medium Priority**: Intimate CareGivers at Elderly’s Home

**Low Priority**: Questionable Events in variable data; WSN Connectivity Errors and Alarms

### 4.2 Functional Requirements traceability

The following below table explains the functional requirements designed to prototype the data decision system on ARM Evaluation Board.

From the sections 1-1, the requirements mentioned are taken into consideration for prototyping. The following below table illustrates the functional requirements of the Smart Universal Gateway and traceability status of each.

<table>
<thead>
<tr>
<th>ID</th>
<th>Functional Requirement</th>
<th>Status</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>UR-1</td>
<td>Acts as bridge between WSN and other various communication networks such as LAN, WAN.</td>
<td>ON</td>
<td>High</td>
</tr>
<tr>
<td>UR-2</td>
<td>Receives sensor data from WSN,</td>
<td>ON</td>
<td>High</td>
</tr>
</tbody>
</table>
4.3 System Requirements Traceability

This section illustrates system requirements designed for the smart universal gateway and traceability of each requirements functions.

The following below are the system functional requirements taken into considered for prototype model of the smart Universal Gateway.

<table>
<thead>
<tr>
<th>ID</th>
<th>System Requirement</th>
<th>Status</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>UR-1</td>
<td>Smart Gateway server should comprise the Data Decision System and reliable database engine.</td>
<td>ON</td>
<td>High</td>
</tr>
<tr>
<td>UR-2</td>
<td>Data Decision System should display real-time activity alarms based on the WSN data from Elderly under surveillance.</td>
<td>ON</td>
<td>High</td>
</tr>
</tbody>
</table>
5 Conclusions and Discussions

In this paper, we propose to exhibit the step by step system development methodology concepts in prototyping the Smart Universal Gateway with Data Decision System implementing the Hidden Markov Model Algorithm for Health Care System. Design and Simulation of the Smart Universal Gateway is done initially and prototyped on ARM Evaluation Board. The resultant is the working ARM board with Decision System integrated with the SQLite database engine. The tasks such as WSN data storage, Decision system implementation is shown.

In future, we aim to design and build the Client and Server applications such as GUI (Graphical User Interface). Following, we may consider Internet Web server and voice call functions in the Smart Gateway. Security issues of the Universal Gateway will also be considered in Future.
6 References


7 Appendix

The following are the topics described in reference to the subjects described in this writing above.

7.1 NFS setup on Host PC

The following files and downloads are done from the repositories of Ubuntu Host PC.

Initially, the below commands are used to install the NFS and Portmap software.

`sudo apt-get install nfs-kernel-server nfs-common portmap`

Configuration of the Portmap is followed and loopback issues can be corrected by editing “/etc/default/portmap” using below commands;

`sudo vi /etc/default/portmap`  or  `sudo dpkg-reconfigure portmap`

Finally, restart Portmap by the command

`sudo /etc/init.d/portmap restart`

7.2 WSN Data Acquisition into ARM Evaluation Board

The following c code file illustrates the functional test of prototyping the Base station data acquisition into the ARM Evaluation Board.

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <errno.h>
#include <sqlite3.h>

int main(int argc, char **argv[])
{
    int fd;
    int i, status, nread;
    char buff[512];
    struct termios opt;
```
// initial declaration of sqlite3 database
sqlite3 *db;
char *zErr;
int rc;
char *sql;
rc = sqlite3_open("mib520.db", &db);
/***********************
// initiate the USB port***************
fd = open("/dev/ttyUSB1", O_RDWR | O_NOCTTY | O_NDELAY);
if (fd == -1)
{ perror("can not open fd\n");
exit(-1);
}
elser
printf ("port open OK \n");
/******************USB setup exception**************
if(tcgetattr(fd, &opt)!=0)
{ perror("Can not setup Serial");
}
/******************setting baudrate*******************/
cfsetspeed(&opt, B57600);
cfsetospeed(&opt, B57600);
opt.c_cflag &= ~CSIZE;
opt.c_cflag |= CS8;
opt.c_cflag &= ~PARENB;
opt.c_iflag &= ~(INPCK | ISTRIP);//
opt.c_cflag &= ~CSTOPB;
//opt.c_iflag |= IGNBRK; //
opt.c_iflag |= IGNPAR; //
opt.c_iflag &= ~(IXON | IXOFF | IXANY); //Disable software control
opt.c_cflag |= (CLOCAL | CREAD);
//opt.c_iflag |= (ICANON | ECHO | ECHOE | ISIG);
opt.c_iflag &= ~(ICANON | ECHO | ECHOE | ISIG); // Choosing Raw input. tcflush(fd, TIOFLUSH);
opt.c_cc[VTIME] = 150;
opt.c_cc[VMIN] = 0;
status = tcsetattr(fd, TCSANOW,&opt);
if (status != 0)
{ perror("tcsetattr fd \n"); // WSN BaseStation Connectivity Notification
}
elser
printf("OK \n");
while(1)
{
  bzero(buff,512);
  while((nread = read(fd, buff, 511)) >0)
  {
    printf("\nLen %d\n",nread);
    buff[nread+1] = '\0';
    for(i=0; i< nread; i++)
    {
      printf("%02hhx ",(unsigned char)buff[i]);
    }
    char data = buff;
  }
  return 0;
}

The Cross compiled binary is moved using Telnet interface and executed in the ARM Evaluation Board.

### 7.3 Sqlite3 Database Compilation & Porting on ARM Board

Compile SQLite using the cross-compiler such as arm-linux-gcc first, get sqlite-3.3.6.tar.gz from [www.sqlite.org](http://www.sqlite.org) unzip it,

```bash
#tar -zxvf sqlite-3.3.6.tar.gz
```

change into the sqlite-3.3.6 directory

```bash
cd sqlite-3.3.6
```

make a new directory such as 'build' under sqlite-3.x.x directory,

open the configure-script using your favorite text-editor ,such as:

```bash
#vi configure
```

I recommend that you make a copy of configure before editing the configure file

```bash
cp configure configure.old
```

and edit the configure.

Comment out the following commands by putting a '#' in front of them(looks like):

```bash
#if test "$cross_compiling" = "yes"; then
# { { echo "$as_me:$LINENO:: error: unable to find a compiler for building build tools" >&5
#echo "$as_me: error: unable to find a compiler for building build tools" >&2;
```
# { (exit 1); exit 1; }; }

#else
# test "$cross_compiling" = yes &&
# { { echo "$as_me:$LINENO:: error: cannot check for file
existence when cross compiling" >&5
#echo "$as_me: error: cannot check for file existence when
cross compiling" >&2; }
# { (exit 1); exit 1; }; }

Save the configure script, change into the build directory you
created and call the edited configure script from the sqlite
directory by using the following option:
../sqlite/configure --disable-tcl --host=arm-linux

after that, configuration should have created a Makefile and a
libtool script in your build directory.
Open the Makefile using your favorite text editor and find the
following lines:
BCC = arm-linux-gcc -g -O2
change to
BCC = gcc -g -O2
The reason for these changes is that the created files have to
be executed on the PC during the compilation, so we have to
compile them with the standard gcc and not the arm-linux-gcc.
if you want compile static library version of sqlite3(only one
execute file for distribution) on ARM, edit Makefile,
find
sqlite3$(TEXE): $(TOP)/src/shell.c .libs/libsqlite3.la sqlite3.h
change to
sqlite3$(TEXE): $(TOP)/src/shell.c .libs/libsqlite3.a sqlite3.h
find
-o $@ $(TOP)/src/shell.c .libs/libsqlite3.la \\ change to
-o $@ $(TOP)/src/shell.c .libs/libsqlite3.a \

save and quit editor
run 'make' command to create the sqlite3 execute file, after a successful compile, Now you should find a hidden "./libs"

directory in your build directory containing sqlite shared object files, like libsqlite.so or static library files like

libsqlite3.a .
run 'arm-linux-strip sqlite3' to decrease the execute file size.
upload the sqlite3 to target ARM9 board by any FTP client and make it executable:
on ARM Evaluation Board with terminal or telnet, run
chmod 775 sqlite3
and then run sqlite3 like this
sqlite3 ex2 or
Enter into the folder location of SQLite3 database and enter the following command
./sqlite3
, if you see the following messages: porting is Successful

SQLite version 3.x.x
Enter ".help" for instructions
title>