Heterogeneous Sensor Networks Design for Aged Care

Heng Tiang Cheng

This report is submitted as partial fulfilment of the requirements for the Honours Programme of the School of Computer Science and Software Engineering, The University of Western Australia, 2006
Abstract

Designing complex applications using wireless sensor networks require solving unique challenges for researchers. The reliability, and accuracy of data collection is important in providing suitable information to care providers. Heterogeneity, and co-relating data from a variety of sources, termed sensor fusion, can provide a mechanism for validating information. Software frameworks need to be created to facilitate these interactions between sensor networks, and the end user.

This dissertation presents a framework for heterogeneous sensor networks to operate. We implement web services to allow sensors to communicate with a database over an IP network. This is very useful for scaling to a large number of distributed sensors, and allows sensor systems to be mobile. Using SOAP allows the database to be stored at an offsite location.

We investigate the capabilities and limitations of RFID, motion detection, and a location tracking systems. We examine the methods of combining sensor data, and evaluate the functional ways to make use of this data. The framework is modified to collect data from these sensors, and process them.

Finally, we discuss software design issues we’ve encountered, and the solutions we used in context for aged care.

**Keywords:** Data collection networks, RFID, motion detection.

**CR Categories:** C.2.4 (Computer-Communication Networks): Distribute Systems - Distributed Applications, C.5(Computer System Implementation)
Acknowledgements

Firstly, I’d like to thank my supervisor Rachel Cardell-Oliver. Her advice, questions and vision gave me direction and focus when I needed it most.

Wavetrend and Aeroscout hardware was provide by Steve van Bloomstein and his team at the Netlink Group. Without their contributions and support, this project would not have been possible.

I wouldn’t have made the distance without the awesome amount of love and support I have received from my Mum and Dad, my brother Heng Teng, my girlfriend Thao, and my best friend Mai.

Many thanks to Sandy Yu who edited my drafts at late hours of the night, making grammatical corrections and identifying my odd obsession with commas.

To everyone else who’s kept me sane and my stress levels tolerable this year. Thank you.
Contents

Abstract ii
Acknowledgements iii

1 Introduction 1
  1.1 Background ......................................................... 1
    1.1.1 Software Engineering Challenges .......................... 1
    1.1.2 Sensor Middleware Issues ................................. 1
    1.1.3 Project Objectives ......................................... 2

2 Background 3
  2.1 Related work ..................................................... 3
    2.1.1 Elder position tracking .................................... 3
    2.1.2 Data Collection in Aged Care .............................. 4
    2.1.3 Information Systems in Aged Care ......................... 4
    2.1.4 Heterogeneous sensors ..................................... 5

3 RFID Systems 8
  3.1 RFID Deployment Scenarios ...................................... 9
    3.1.1 Mobile Tag with Static Location Readers ................. 9
    3.1.2 Mobile Reader with Static Location Tags ............... 9
    3.1.3 Mobile Readers with Mobile Tags ......................... 10
  3.2 RFID Pilot Implementation and Evaluation ..................... 10
    3.2.1 Evaluation of Wavetrend RFID System .................... 10
    3.2.2 Factors in Relative Signal Strength Indication values.. 11
    3.2.3 Evaluation of RSSI and Distance ......................... 11
    3.2.4 Radio Grey Areas ........................................... 11
A  Original Honours Proposal  28
B  Motion Detection Algorithm  33
C  SOAP Message Contents  34
List of Tables

3.1 Table of RFID reception based on Antenna type . . . . . . . . . . 10
List of Figures

3.1 Basic RFID System ................................................. 9
3.2 Range of RSSI Values Over Distance ......................... 12
3.3 Individual Range of RSSI Values Over Distance - Parallel to Reader 12
3.4 Individual Range of RSSI Values Over Distance - Across the Reader 13
5.1 Using Reference Tags Improve Positioning Accuracy ........... 19
CHAPTER 1

Introduction

1.1 Background

With the increasing availability of low-cost, wireless microsensing devices and advancements in pervasive technologies, there is growing demand in deploying these devices in the healthcare industry to meet demands for care provision [9]. The aged care industry is a specific aspect in healthcare, with unique requirements, such as long term care and monitoring of elderly patients.

Pervasive technologies in high-care facilities allow nurses to provide more efficient care to their patients, and meet the challenge of increasing demand for aged care. Often when an elderly is admitted to hospital, their health declines from being out of their normal environment. We can use pervasive technology to provide intelligent environments for supporting the elderly for independent living.

1.1.1 Software Engineering Challenges

Applications written for the healthcare domain pose unique software engineering challenges. Healthcare applications are safety critical systems and the failure of an application to perform to specification, or to anticipate exceptional behavior can lead to loss of life such as the Therac-25 accident [15]. There is no simple solution for this problem [11], so implementations must find ways to display a high level of reliability and accuracy in information systems.

1.1.2 Sensor Middleware Issues

The technologies that track and monitor patients have been available for a long time. However, efforts at integrating these systems into IP-based networks has been lacking, and state of the art research into sensor middleware is still in the
early stages [8]. There are open questions on how to manage sensor resources, scalability, mobility, heterogeneity, real world integration, specific application design, data aggregation, quality of service, and security.

1.1.3 Project Objectives

The objective is to use different types of sensors to accurately detect and gain useful information on an event. Project time constraints allow for a small selection of events, but the approach must be scalable to larger projects.

Chapter 2 provides a review of research topics relevant to this paper. We explore a broad range of topics on software component engineering, data collection networks in general and medical context, and heterogeneous sensors. We review number of wireless positioning systems, and explore projects using signal strengths to triangulate transmitter positions.

Chapter 3 explains RFID systems in general, and the evaluation of the Wave-trend RFID system. We describe the design and implementation of this system.

Chapter 4 covers visual motion detection sensors and the evaluation of a webcam motion sensor. We measure the effectiveness of coverage and accurate sensing. From these results, we present recommendations on how to deploy a web cam for motion sensing.

Chapter 5 covers wireless location based systems and the operation of the Aeroscout 802.11 wireless location system. We investigate the accuracy of the positioning system with, and without reference tags.

Chapter 6 describes a number of the design issues we encountered in creating a heterogeneous sensor network, and our proposed solutions. We build a software framework on Web Services, and discuss our methods for resolving these design issues.

Finally, we present our concluding remarks in Chapter 7. We discuss the important aspects of the project, and make suggestions for future work.
CHAPTER 2

Background

A project of this nature is broad and encompasses many research areas of computer science. A sample of papers on the relevant topics are presented here.

2.1 Related work

2.1.1 Elder position tracking

The House of Matilda location position system [18] allows proactive, context aware services to meet the functional goals of allowing an elder to continue independent life. Their requirements for a location system were low cost, high accuracy, portable, and low power consumption.

S. Helal [10] et al differentiate between location tracking systems that allow a central computer to continuously keep track of user locations, and location positioning systems that receive location information from the environment and calculate their own position. They implement an ultrasonic sensor system to calculate the location of an elderly patient.

Ultrasonic positioning systems use the transmitted signals time of arrival to calculate distance. The receivers are positioned at 4 corners of a rectangular room. To successfully infer the patients location, 2 or more TOA signals are aggregated, and a simple geometry algorithm used to triangulate the location. An Open Services Gateway Initiative (OSGi) interface allows services to access the location data through web services.

Wireless location systems are vulnerable to interference [4]. The project classified errors into 4 categories hardware errors, background noise, receiver synchronization errors, and reflective errors. Experiments measured the error rate to be 3.3%.
2.1.2 Data Collection in Aged Care

In 2004, Behavioral Informatics Inc performed a pilot study to evaluate a method for continuously, and unobtrusively monitor the daily activities of people requiring a high level of medical care [7].

They employed wireless motion detectors within the living environment such as the bathroom as well on significant objects such as the fridge. A base station collected the data from all the sensors, and forwarded it to a web server in XML format. A reporting system analysed the action trends, and gave periodic updates on the clients activity levels and infers their status of health. Data from this was collated into an information pyramid.

They present a 4 layer pyramid information model to describe the processes from data collection to intervention. This information model is a good framework for establishing sensing services. There are 2 categories of information. Timely status information is for daily events, such as bathing, taking medication or preparing a meal. Crisis events such as a fall in the bathroom can also be inferred from status information.

Trend analysis of status information is used for medium to long term care planning. This analysis provides care givers with advice on the activity levels of their patients. Advice is presented in a simple, clear user interface. Important trends are represented through simple colored icons. Importantly, trend analysis can provide advice on longer term concerns such as nutrition, activity levels, and long term health. The care giver can refer to the trend analysis to gauge the effectiveness of their intervention, and decide if further action needs to be taken to ensure the safety of the ward.

There is a correlation between accuracy of data at the collection layer, with accuracy of data at higher layers in the information pyramid. The more accurate the data collected, the more effective the care plan can be. Conversely, the less accurate the data collected, the less effective the care plan, and in certain cases, it can have negative effects on the patient.

2.1.3 Information Systems in Aged Care

Georgia Institute of Technology are investigating the usability issues in classifying, and presenting medical status to the care network of the elderly [18]. They present their work on 3 areas of computer science; Human-Computer Interaction, computational perception, and cognitive aging. Similar to the Hourglass [19] project, they recognise that the solution requires a multidisciplinary approach. Their goals are to integrate computers as a pervasive enhancement to an elder’s
life, develop intelligent sensors that can interact with people, and design interfaces that support daily life for an elder without being complex or obtrusive.

The researchers classify information into 3 categories. Activities of daily living are basic functions performed by an elder such as eating and bathroom activities. Instrumental activities of daily living are define as acts that involve interaction with an elder’s everyday environment, for example cleaning and taking medicine. Enhanced activities of daily living is described as acts that involve the elder’s interaction with the outside world, such as communicating with family and friends.

From these information measurements, they can ascertain an elder’s health, and monitor for abnormal behaviour. The information provides trends and awareness of an elder’s short and longterm health. This information needs to be presented to care providers such as nurses and family members for any remedial action.

The Intel Computer-Supported Co-ordinated Care project is a project with similar outcomes [5]. They conducted face to face interviews with elders, and their care network in order to measure the relationships that care providers have with their elders, and the impact on daily life for the carers. They use metrics on distance from the elder’s home, the flow of communication with elders and other members of the network and the roles and responsibilities of each carer.

From this extensive requirements elicitation process, they found that some elders have primary and secondary carers. Members can move from a secondary role to a primary carer, with a drastic impact on the carer’s life. Information dissemination between carers within an elder’s network is a difficult process. Elder’s needs are also changing, and carers are not always aware of these changes.

They outline 3 characteristics of a suitable solution. Focus on the person, and consider issues of privacy, trust and emotion. Organisational structure of networks vary for each individual, and over time. Finally, the process of care provision should be in the background, and not the focus. Information systems should augment current processes, instead of creating new ways for carers to work.

2.1.4 Heterogeneous sensors

Marques et al at Florida Atlantic University have implemented a heterogeneous sensor network to study the integration of location tracking with video analysis in providing a safe environment for the elderly [17]. The platform is based on a medical sensor network Codeblue, and is extended using RFID sensors, and a video monitoring system. The object of this project is to detect when a patient
has moved into a restricted area, and to detect a crisis situation through unusual patient behavior.

The project explores the use of heterogenous sensors to detect when a patient leaves their room. Video cameras are programmed to monitor a section of the room, and will create an event when the patient has moved away from their room. Initial results show that the camera system is able to accurately detect patient movement from their room to a restricted area. The RFID sensors also provide redundant location information on the patient for verification of the camera information. Future work is directed at performing image analysis to detect a patients status and if they require crisis attention.

Hourglass [13, 19]is a Harvard University project that addresses recent challenges and issues in sensor networking. Their goal is to build a scalable, robust infrastructure to discover, query and deliver sensor network data. Their design allows applications to harness the functionality of many disparate sensor networks and infrastructure-based services.

To meet these goals, they present a state of the art solution for the challenges of intermittent connectivity, resource identification and discovery, service composition and heterogeneity. They use a virtual circuit model where sensor data sources are connected to receivers through network links. These network links are not dependent on a single protocol and can be TCP/IP or other network protocols. This allows developers to write custom routing algorithms and build network level services such as buffering and compression. Additionally, data can be processed throughout the circuit. Individual data and network nodes can filter, aggregate, compress and buffer data based on environmental factors and the data network design.

Control channels are built into the circuits, allowing queries to be constructed and the level of autonomy and processing for sensor networks to be specified. The service infrastructure model provides applications with a known interface for accessing a circuit and gaining access to the data collection network. Services are listed and described in a registry. An application can be composed of a number of services and take advantage of multiple data collection networks. This design allows the sensor layer to be separated from the network layer.

Data aggregation, trend analysis and event triggers from multiple types of sensors provide a more detailed view of a monitored system. For example a location sensor will detect when a patient is in a dangerous area of the care facility. If a patient collapses, video analysis will detect that the patient has not moved. This triggers video analysis of the room at faster intervals to detect if the patient has fallen over.
Node heterogeneity is discussed in terms of communication reliability. Ad-hoc sensor networks often have a core node or base station that has stable communications properties reference. These core nodes can provide storages and network links to periphery nodes, allowing data to transverse to a central point for data collection.

Heterogeneity is discussed in terms of different node properties within a single type of sensor network. Node heterogeneity provides reliable and robust routing services by utilizing stable core nodes for aggregation and data storage. Periphery wireless nodes have unreliable network links and require buffering and robust routing algorithms.
CHAPTER 3

RFID Systems

RFID stands for Radio Frequency Identification and is a method of identifying objects by sending electronic data over a wireless medium. RFID systems consist of tags, readers and processing software. An RFID tag generates electronic data and transmits it through a radio signal. RFID readers pick up this signal and measure the Relative Signal Strength (RSSI) and forward the data through a serial port to processing software. Software is required to capture, aggregate, filter and store the received data.

Active RFID systems have an internal power supply, which gives the tag a range up to 100m. Data packets are generated every couple of seconds. The packets are received and decoded by a RFID reader and forwarded to a computer for processing. The software needs to process all these signals by filtering and aggregation to recognize the events that have occurred. Active RFID tags are more expensive to manufacture than passive RFID tags. Their capabilities are more useful than passive RFID systems. Active RFID tags can carry additional sensor such as motion, or temperature. The data from these sensors are included with each radio packet.

Passive RFID systems require the tag to be powered either through the small amount of current in the received signal, or generated through magnetic induction, by moving the tag through a magnetic field. When tag is close enough to a power source to receive power, it is activated and sends data. Passive RFID tags are very cheap to make and are generally used for a single cycle before being discarded.

One of the earliest RFID systems is the secondary surveillance radar systems used to identify passenger aircraft on air traffic control radars. Other systems are used in asset tracking, building access control and process automation. The technology is commonplace that many people carry RFID transponders and use the technology on a daily basis. UWA student cards are passive RFID tags. Remote car alarm systems are an example of an active RFID system.
3.1 RFID Deployment Scenarios

When designing RFID systems, it is important to consider a number of variables. RFID systems are imprecise positioning systems. RFID readers have an area of coverage, depending on its specified sensitivity and antenna used. The coverage zone indicates that a tag is close to a reader and is estimated as a circular area, with the reader at the center. Reader coverage zones can overlap, providing full coverage, or can be independent. Overlapping zones require additional processing to compare a radio link metric such as RSSI to determine which reader is the closest to the tag. Independent zones will indicate that a tag has moved through its location. The tag has to pass through 2 readers to determine its direction Figure 3.1.

There are 3 common scenarios in deploying RFID systems.

3.1.1 Mobile Tag with Static Location Readers

The RFID tag is attached to the mobile object being tracked. The RFID readers are positioned in known, fixed locations. As the object moves around, it transmits its wireless signal to all readers within range. This allows the centralized tracking system to be aware of the tags location and hence that of the mobile objects location.

3.1.2 Mobile Reader with Static Location Tags

The RFID reader is attached to a mobile device, usually carried by a user. RFID tags are positioned in known fixed locations. The data processing is done on
the mobile device to determine its approximate location based on the tag IDs it receives.

3.1.3 Mobile Readers with Mobile Tags

RFID readers are attached to mobile devices. Mobile tags are attached to assets that are to be identified. There is a high level of mobility in this scenario, so transmission range is kept to a low level, under 5 meters. This is useful in the event of the coverage needs to be mobile.

3.2 RFID Pilot Implementation and Evaluation

3.2.1 Evaluation of Wavetrend RFID System

I obtain a basic RFID kit which consist of a Wavetrend L-RX201 RFID Reader, two L-TG 501 RFID Tags and the Wavetrend Software Development Kit version 1.0 [3].

L-RX201 RFID Reader is a 433Mhz receiver that is rated to -103dBm sensitivity, at 700kHz bandwidth. It connects to a PC via the Serial RS232 protocol and is capable of data transfer speeds of 9600kbps up to 115200kbps. Minimum RSSI is 130 and Maximum RSSI is 70. The stub antenna is used for the pilot project, radio signal reception is dependent on the antenna used. Table 3.1 lists the range for the L-RX201 Reader.

<table>
<thead>
<tr>
<th>Antenna Type</th>
<th>Minimum Range (m)</th>
<th>Maximum Range (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0.2</td>
<td>7</td>
</tr>
<tr>
<td>L-AN200 (Stub)</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>L-AN100 (Whip)</td>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>L-AN300 (Patch)</td>
<td>3</td>
<td>100</td>
</tr>
</tbody>
</table>

TG 501 RFID Tag is a 433Mhz Transceiver designed for personnel and asset tracking. The retransmission time is programmable and is set to 3 seconds. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is implemented, so retransmission times are random. Surface Acoustic Wave (SAW) is implemented to provide accurate radio signal transmission. Wavetrend tags can also detect movement. The tag contains a small mercury switch, which flips when the tag moves. When the switch is triggered, a custom integer field is incremented
on the tags internal memory. This field is included with every data packet sent. The software will compare the current integer with the previous integer and if there is a change, it will call the onTagMovement event.

Wavetrend Software Development Kit is available as an ActiveX control for Visual Basic 6.0. It uses a hierarchy of objects to store information about the system components such as the Readers and Tags. Wavetrend RFID systems can be deployed up to 255 readers to a single Serial connection and support a large number of Tags.

3.2.2 Factors in Relative Signal Strength Indication values

RSSI is a measure of signal power over distance and obstacles in the signals path and is an indication of signal quality.

3.2.3 Evaluation of RSSI and Distance

We measure the relationship between RSSI values and distance. We place 2 tags at 1m intervals in a room and record the RSSI values over a 5 minute period. The average values are plotted and the max and minimum values are also recorded. Over some distances, the range of maximum and minimum values is very close to each other. Other measurements show a high variability of values. A possible explanation could be multipath, where the signals are reflected and weakened before reaching the RFID reader. There does not seem to be a direct relationship between distance and RSSI variance, at distances under 8 metres.

3.2.4 Radio Grey Areas

We measure the relationship between RSSI and distance. We place 2 tags at 50 cm intervals in a room and record the RSSI values over a 2 minute period. The average values are plotted and the graphs show that RSSI values can vary significantly over large values over the short distances.

Jerry Zhao and Ramesh Govindan at University of Southern California explored the performance of packet delivery in wireless networks [20]. They found there are areas within a radio receiver’s coverage that have unreliable packet delivery. These grey areas are within moderate range of the receiver, not at the edge as previously assumed. Our results reflect this in Figure 3.3. We rotate the RFID Tag 90 degrees to be across the reader and repeat the experiment. The signal strength is more stable, with an acceptable variance of 3-10 RSSI. There is
Figure 3.2: Range of RSSI Values Over Distance

Figure 3.3: Individual Range of RSSI Values Over Distance - Parallel to Reader
an anomalous reading in Figure 3.4(d), which is probably random radio interference affecting a single packet.

3.2.5 Conclusion

Orientation of the tags to the antennas affects RSSI values. This is the easiest way to modify RSSI values. There is a difference of up to 20 RSSI when a tag is rotated 90 degrees, at the same distance from the reader. The human body is also a source of interference. Having the Tag and Reader 50cm off the floor helps improve radio reception. These results show that a positioning system based on Wavetrend technology would be extremely inaccurate. By rotating a tag 90 degrees, the signal strength can be changed significantly. A similar project using the same hardware [6] also confirms these findings. The optimal solution is to use the readers as zones within a building, so the position of the tag is known to be within an approximate area. Exact position of an object must be obtained by other methods.
CHAPTER 4

Motion Detection Sensor

Real-time visual patient monitoring provides useful information to aged care providers. Using simple motion detection algorithms and human modeling, it is possible to detect events such as a person falling down. Sensing human movement through computer processing is a complex task. Motion capture algorithms for surveillance have unique models for analysing body motion. The functional structure of a motion detection sensor would be tracking, estimate position and recognition. Tracking motion involves separating the moving object from the static background. Human models are used for estimating the pose of the object for example standing, sitting, or lying down. The sensors can recognize situations based on the estimated position for example a person has fallen down and is not moving. This chapter evaluates a simple algorithm for sensing human movement.

I use a simple pixel by pixel comparison based on greyscale intensity values. The sensor initializes and captures an image of the background. This image is then subtracted from the next frame, pixel by pixel. When there is a difference between the background pixel and the current pixel, it is recorded as a possible movement. Depending on the threshold set, if enough pixels are recorded, then the system recognizes that there has been a change in movement. A bounding box is drawn around any area of changed pixels and an event is triggered [12]. This bounding box can be used for analysis of human position, such as sitting, standing, walking and lying down. When the motion alarm is triggered, it stores the current frame and date-time into a remote database using a web service. The sensor then pauses for the next 5 frames to save on processing and avoid flooding the database.

4.1 Motion Detection Sensor Evaluation

We investigate the coverage and accuracy of the sensor at detecting movement in a computer science lab. The area is cleared of obstacles and performed using a single moving person. We use a simple webcam, resolution is at 640 by 320,
with a manually adjusted focus lens.

4.1.1 Coverage

Experiments are performed on camera direction and height to determine the largest possible area of coverage. The room is divided into 1m x 1m squares and a person is directed to walk from one side of the room to the other, until she has traveled through all the squares. This indicates how much of the room is visible by the camera. We found that a single camera can not able to provide 100% coverage. The optimal case is when the camera is positioned to cover areas of high traffic. This gives the sensor the best chance of detecting motion. Alternatively, the sensor can be placed over entry and exit points of a room, to detect when someone has moved into and out of the area. This has the advantage of knowing that someone is, or is not in a room.

4.1.2 Accuracy

We experiment with lighting conditions to find out how effective the sensor at detecting motion in less than ideal conditions. Each test was repeated 3 times and the results recorded. Camera position was the important variable in motion detection. The height of the camera and the direction dictates the coverage of the room and how accurately motion can be detected. We positioned the camera using the results from the coverage experiment to cover the high traffic areas.

In daylight conditions, the motion detection sensor is able to differentiate between the subject and the background. A clear image is captured and the outline of the person is drawn on the frame. The sensor is capable at finding the human model, but can be confused by shadows and fast changing ambient light. This information is too unreliable to be analyzed automatically. The motion detection software is modified to store the current frame if motion is detected.

In low light situations, the motion detection sensor is unable to differentiate between the subject and the background. Decreasing the pixel threshold increases the sensitivity to detecting movement in these conditions. However, the image quality is poor and a human model of the motion is hard to present. Further image analysis of human position will be difficult to perform in this situation by a human operator or through software.
4.1.3 Conclusion

Optimal effectiveness for motion detection is when the camera is positioned to cover areas of high traffic, with good room lighting. The motion detection algorithm is capable of detecting motion, but can be confused by shadows, or areas of small movement.
CHAPTER 5

Wireless Positioning Systems

5.1 Aeroscout Location System

The Aeroscout location system [1] uses 802.11G access points to triangulate the position of RFID tags. The positioning engine can be software or hardware based. Aeroscout takes RSSI values from each access point and calculates a relative position on a map. The system has setup, calibration and operation phases.

5.1.1 Setup Phase

The setup phase involves setting up 802.11G Cisco access points and configuring them to listen for wireless multicast packets on a particular channel. A minimum of three access points are required to provide positioning services through triangulation. It is important to follow 802.11 best practices to optimize packet reception. Each access point is given a wireless channel as far from its neighbour to prevent transmission and reception interference. The dual antennas are configured for diversity operation.

5.1.2 Calibration Phase

The calibration process provides accurate location positioning. A building map is imported into the system. The map is scaled and an origin is selected. Object locations are given as (x,y) co-ordinates relative to the origin. The access points are placed on the map at (x,y) co-ordinates. Other static objects such as reference tags are also positioned manually on the map.

Static tags are also positioned in every room. The positioning software uses these reference tags as reference RSSI points. The reference tags allow the positioning engine to know RSSI values at known locations. This is useful to adjust
for signal degradation by walls, obstacles and sources of interference. Signal degradation will cause a tag to appear further away than it really is because the signal has been reduced by factors other than distance. These reference tags provide a real-time calibration of the software. A problem with calibration, is the requirement to recalibrate when the sensing environment changes. This can be from moving furniture in a room, to insulating walls. MoteTrack [16] calibration took 5 days and 500 signatures to collect. Using ongoing, real-time calibration avoids the time cost of having to physically recalibrate the RSSI database. The disadvantage of using reference tags is having a large number of tags that are not used for tracking and taking up wireless bandwidth.

5.1.3 Operational Phase

The positioning engine tracks all tags within range in real-time. Tags transmit at 30-second intervals to conserve power consumption. Increasing the transmission time will increase the response of the system at tracking tags, but will reduce the battery life.

Each Aeroscout tag is a 802.11 transmitter and has a globally unique MAC address for identification. IEEE 802.11 specification describe 11 channels for clear data transmission in the 2.4Ghz range. The tags can transmit on three of these channels, so the access points need to be configured for this. Channels 1, 6 and 11 are selected as they are the most separate from each other and will cause the least interference.

5.1.4 Aeroscout Evaluation

We evaluate the effectiveness of using reference tags in calculating accurate positioning. Five access points are installed at the corners of the building providing full, overlapping wireless coverage. Pairs of tags are positioned in different rooms. One tag is configured as a reference tag. The other tag is placed a metre from the reference tag. The first measurements use RSSI calculation only and the second measurements include reference tag adjustments.

Reference tags improve the precision and accuracy of the positioning engine. The tags were consistently within a 2 metre radius of the reference tags in Figure 5.1(b). Without the reference tags, the positioning engine was inaccurate and would often show the tags moving, or positioned up to 10 metres away from its actual location Figure 5.1(a).
5.1.5 Using TDOA

Using the hardware based positioning engine would allow location calculations based on signal Time Difference of Arrival (TDOA). TDOA uses the known property of the speed of the transmission to estimate the approximate distance from the receiver. The tags and receivers are time synchronized during calibration. When the tag sends a data packet, it timestamps the packet. The receiver will compare the timestamp with its current time to establish how long the packet took to send and therefore the distance to the tag is. This is a more accurate method of position calculation because it is not badly affected by sources of signal strength interference. This would improve the accuracy and precision of the system. The challenge is making sure the tags and readers are synchronized.

5.1.6 Using Exciters

802.11 wireless works in a 3D environment. Buildings with multiple stories would find it difficult to tell which floor the tag is. The positioning engine is only capable of 2D co-ordinates (x,y). Tags can be tracked using exciters. When a tag is in range of an exciter, it transmits its normal data packet at faster intervals. The exciter will also include a message for the tag to send with its packet. These messages can be customized, such as a floor co-ordinate. Exciters can be positioned at the entrance to every level and when a tag goes through the exciter, the tag will include the message that it is on floor X. The positioning engine can then use the information to correct place the tag on the correct floor, as well as the X,Y co-ordinate.
CHAPTER 6

Heterogenous Sensor Network Design

6.1 Web Services

Web services is a modular framework for interprocess communication. Its a standard means of interoperation between different software applications, running on a variety of platforms and/or frameworks. It provides a layer of connectivity independent of the programming language used by the communicating software. The term web service originates from the method of communication, which is through web-based transport protocols such as HTTP, SMTP and FTP. There are 4 generic layers to Web Services.

The service discovery layer is the first step towards using a web service. The Universal Description, Discovery and Integration protocol (UDDI) is a mechanism for querying a server for the services it provides.

The service description layer describes the service at a programmatic level. The Web Service Description Language (WSDL) is a standard to share this data between applications. It is a structured XML file that defines the public methods, parameters and return values, including the data-types of a service. This allows requesting applications to know how to call a method of a remote application. The WSDL is normally used at development time for the developer to make use of the available service providers on the network.

The messaging layer uses SOAP and is responsible for encoding data into XML format that is understood by both service provider and requester. Request and response messages are sent in a SOAP envelope. The service is accessed through a Universal Resource Identifier (URI) in the header of the SOAP message.

Finally, the Service Transport layer is the method for transporting SOAP messages. Common web protocols such as HTTP, SMTP, or FTP can be used. Developers can implement a simple listener within the application and capture the network stream and decode the stream into SOAP and respond via the same method. Alternatively, web services can run from a web server, through Java
Server Pages, or Microsoft ASP.NET.

SOAP stands for Simple Object Access Protocol. A SOAP message is the core of a web service. It specifies parameters and the method and method that is to be used. SOAP messages can also contain security mechanisms such as encryption and authentication. The SOAP message to insert a line of RFID data into a database is 598 bytes of overhead, to send 3 integers and a string. Multiply that by 40 packets a second and that is a lot of bandwidth for a single tag. The software filters the data to only sending messages when important events occur.

6.1.1 Loose Coupling

Loose coupling is a term for describing a robust relationship by 2 systems for exchanging data. If a requesting agent changes the elements within its messaging format, the service provider is still able to understand the message and respond in a correct way. This reduces the risk that a change in one part of a software application will have flow on effects on dependent software modules. Web services allow software to interoperate loosely by adding a communication layer that abstracts any programming language complexity.

Software components are an example of loosely coupled. Components are stand alone software libraries that provide a service for an application. We used the Microsoft Web Services software component in the Wavetrend software because the development environment did not natively support Web Services. As well as providing the functionality of normal

6.1.2 Choosing a Development Environment for Web Services

I considered J2EE and .Net platforms for the development of the database web services. Either platform would be suited, with strong support for Web Services through their toolkits.

J2EE is a well supported Java development environment, with a large open-source community for Web Services. Frameworks such as OSGi and Glassfish are strong tools used in the development of Web Services. However, Java is a complex environment for development and has a high learning curve.

I chose Microsoft’s Visual Studio 2005 for its development simplicity and powerful Web Services libraries. Software libraries from Microsoft are released with good documentation and sample code. Creating a web services method is simply appending a [WebMethod] over the methods header. Additional options such as
authentication can also be included in this way. The WSDL files are automatically created at compile time. The auto-generated code for database access is a huge saving on development time. Software components such as Web Services are easily imported through the Web Reference interface.

Open Services Gateway Initiative Platform is a framework for developing and deploying software components for an application. Software components can interact and collaborate with each other through the OSGi framework. OSGi has been popular with the pervasive computing community with how software components are easily integrated [14] The problem with OSGi is its dependency on software written in Java and running on the Java virtual machine.

6.1.3 Project Implementation

This web application can be broken down into 3 layers. A HTTP server provides an interface for web service clients to query. I implement Microsots Internet Information Services 6.0 and ASP.Net 2.0. A business logic layer is for the filtering, aggregation and processing of data received. Microsoft Visual C# is chosen for its rapid application development and the supported libraries for Web Services and Database access. Finally, the persistence layer is using Microsoft SQL Server Express. SQL Server Express is a lightweight version of the enterprise SQL Server 2005, suitable for basic web development. It is a relational database and is built into the Microsoft Visual Studio 2005 development environment. These tools are provided free by Microsoft.

These 3 layers are loosely coupled. Installing upgraded patches for IIS, or SQL Server Express will not affect the business logic layer. A different database can be used, as long as the data dictionary is kept consistent. The connection string may need to be changed, but the database TSQL commands are auto-generated at development time and can be easily changed to support other relational databases. This level of coupling is suitable for this project.

Sensor software is written in many diverse languages. For this project, RFID software was written in Visual Basic 6.0. The motion detection software was written in Visual C# 2005. Alternative motion detection software was written in Java using the Lejos Vision API [2], but was not integrated into the system. The database backend runs on a SQL Server Express server and Microsoft IIS 6.0. Distribution systems were a design required, where sensors are controlled by individual computers. A standardized, interoperation communications protocol is required for these sensors to store and share data. Web Services was chosen as the communication stack for its vendor support, interoperability and implementation simplicity.
6.1.4 Database Web Services

The database is the core of this system. All sensors interact with the database through individual, customized public interfaces as a web service. A table for each sensor is created, with columns for individual data elements. I implemented two interfaces for storing data, one for Motion Detection sensor and one for RFID sensor. In addition, I implemented two interfaces for querying the database for data from the respective sensors.

An IP-based network is implemented, to allow distributed sensors to communicate with the database. Using IP allows communication over a variety of transmission mediums such as Ethernet, GPRS, or Satellite.

SOAP communication has a high overhead from the HTTP handshake, SOAP header and metadata associated with each message transmitted. Data processing should be done as close to the source as possible for timely processing and to reduce SOAP traffic across the network.

The RFID software was modified to send SOAP messages to the database. Microsoft SOAP library Version 3.0 was used to simplify development. When a tag is in range of a reader, the tag ID, RSSI, packet number and date received is sent to the remote database for storage and analysis. Other events such as when a tag moves to another reader, the tag times out, or a new tag is found are also sent to the database by SOAP.

6.1.5 Conclusion

Both Java and .Net Web Services toolkits are suitable for middleware development in this project. The .Net toolkit was chosen, because it was simpler to install and use, having an all in one development and implementation environment and an easy to use user interface. The auto-generated code is also a timesaving feature that is used throughout the project.

6.2 Remote Processing against Local Processing

The motion detection sensor captures data at 15 Frames per second. The SOAP overhead to send each frame is X bytes. The processing system would require Y bandwidth to receive these frames. In addition, network latency would decrease processing further. When the sensor is modified to send a message only when motion is detected, SOAP traffic slows down to once every 5 seconds at the worst case. Information is stored in the database in a timely and more efficient manner.
6.3 Heterogenous Sensor Real-Time Information models

Real-time information queries is an important element of a heterogenous sensor network for aged care. It is critical that important information is presented to care staff as soon as possible. These models were implemented in 2 software pilot front-ends. The data backend is kept consistent. The information models are implemented at a business layer, consisting of simple rules for verifying and aggregating data from the database. There are 2 information models in this implementation.

6.3.1 Verification Model

Each sensor data value is compared with the others and an event is verified according to a set of rules and/or weighting. This increases the accuracy of information collected. When data from 2 or more sensors have conflicting information, it can identify false positives from one of the sensors.

We used a Wavetrend TG800 tag and a motion detection camera system. The TG800 has an internal motion detection sensor which detects if the tag is in motion. The motion detection camera will detect motion within its field of vision and draw a bounding box around the area of movement.

When either sensor detects motion within its sensing area, it sends the data to the web service database. If only one sensor has detected motion, the reporting interface will report a possibility of movement in the room. When both sensors have detected motion within a minute of each other, the reporting interface will confirm that there is movement within the room.

6.3.2 Aggregation Model

Sensor data is accumulated and added together. The sum of the values is then used in information models to present more detailed information on the patients situation. The Aeroscout positioning system is unable to calculate when a patient is moving until a few minutes after they have moved. If they are constantly moving, an additional sensor can be used to track this information. We use a motion detection Wavetrend RFID tag for this purpose.
6.4 Sensor Management

Sensors need to be set up, configured, calibrated and managed during their operation. This design does not support dynamic modification of sensor operation at runtime. Although it would increase usability to have a single location to configure and manage all the sensor networks, it would add a layer of complexity that is not necessary. The positioning sensor has support for remote management through a web browser. The motion detection sensor and the RFID sensor do not support remote management natively, so this would be additional functionality to add. In addition, the design rationale is that each sensor is responsible for its own operation, to maintain a loose coupling between the core service and the sensors.
CHAPTER 7

Conclusion

We have proposed and implemented a web services framework for attaching heterogeneous sensors. Distributed sensors communicate with a remote database using SOAP over a HTTP transport protocol. The system can be easily modified to additional sensor types and collect different types of data as requirements change. The sensor software was easily modified using web services software components to reduce the development time required to create a functional proof of concept.

Sensor capabilities were examined and integrated into a system for tracking an elderly patient and detecting their movements. Data collected from heterogeneous sources can be aggregated and verified using data from each other using different types of sensors.

7.1 Future Work

Heterogenous sensor network design for aged care applications must address the chalanges of data collection networks within the relevent application space. While this project has approached four of the issues, the problem domain remains very large.

It will be very interesting to develop the reporting side of the system in the same way. As the system scales to a large number of sensor networks and data collection increases significantly, AI agents will prove very useful in sorting the data and finding critical events. In addition, intelligent fault tolerance and failover can be added.

Implementing a dynamic reconfiguration component would make sensor integration to be much easier. This component would take inputs on the data to be received by the sensor, and dynamically create the web service for that sensor.

The initial proposal included an investigation into the security, and privacy issues in pervasive technologies. Time constraints made this a low priority. Future
work in this application space should consider the impact on the elder’s privacy, and using security measures
APPENDIX A

Original Honours Proposal

proposal.pdf
=210mm =297mm
APPENDIX B

Motion Detection Algorithm

If backgroundFrame == null
   backgroundFrame = image;
else
   currentFrame = image;
For every pixel in currentFrame
Do a comparison with the same pixel in the backgroundFrame
If different pixel
   Increment pixel count
   Set tempFrame value = Black
If same pixel
   Set tempFrame value = White
Detect the edges of the black blob
Write the edges of the blob as a red box around the currentFrame
APPENDIX C

SOAP Message Contents

The HTTP command:

POST /DataStore/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/RFIDAddtoDB"

The SOAP message is then sent

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<RFIDAddtoDB xmlns="http://tempuri.org/">
<tagID>int</tagID>
<tagAge>int</tagAge>
<rssi>int</rssi>
<TimeRecorded>string</TimeRecorded>
</RFIDAddtoDB>
</soap:Body>
</soap:Envelope>

The Web Service then returns a message to confirm the data has been received.

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <RFIDAddtoDBResponse xmlns="http://tempuri.org/">
            <RFIDAddtoDBResult>boolean</RFIDAddtoDBResult>
        </RFIDAddtoDBResponse>
    </soap:Body>
</soap:Envelope>

This code is kept within a try/catch block. If there is a failure to send the message, such as network failure, the software can catch the exception and perform recovery actions such as buffering the data for sending later.


