Context-Sensitive Quality Data Management for Pervasive Computing Environments

Dissertation submitted to the National University of Ireland, Cork

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy of Computer Science

October 2007

John Martin O'Donoghue B.Sc. M.Sc.

Department of Computer Science

University College Cork

Ireland

ABSTRACT

Pervasive environments generate large quantities of data, originating from backend servers, portable devices and wireless mobile sensors. Pervasive sensing devices that monitor properties of the environment (including human beings) can produce a large data source. The unprocessed datasets may include data that is faulty and irrelevant, and data that is important and useful. If not managed correctly the large amount of data from a data-rich pervasive environment may result in information overload or delivery of incorrect information.

Context-sensitive quality data management aims to gather, verify, process, and manage the multiple data sources in a pervasive environment in order to deliver high quality, relevant information to the end user. Managing the quality of data from different sources, correlating related data and making use of context are all essential in providing end users with accurate and meaningful data in real-time. This requirement is especially true for critical applications such as in a medical environment.

This thesis presents the Data Management System (DMS) architecture. It is designed to deliver a *quality of data* service to its users. The DMS architecture employs an agent based middleware to intelligently and effectively manage all pervasive data sources, and to make use of context to deliver relevant information to the end-user. DMS components have been designed to manage: data validation; data consistency; context-based data delivery; knowledge management and distributed processing. The DMS components have been rigorously evaluated using various medical-based test cases.

This thesis demonstrates a careful, precise approach to data based on the quality of the data and the context of its use. It emphasises the DMS architecture and the role of software agents in providing quality data management.

DECLARATION

This dissertation is submitted to University College Cork, in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Science. The research and thesis presented in this dissertation are entirely my own work and have not been submitted to any other university or higher education institution, or for any other academic award in this university. Where use has been made of other people's work, it has been fully acknowledged and referenced.

Excerpts of this thesis have been published in journals, conference/workshop papers and posters, namely:

[O'Flynn,07],[Angove,07],[O'Donoghue,06a],[O'Donoghue,06b],[O'Donoghue,06c],[O'Donoghue,06d],[O'Donoghue,06e],[O'Donoghue,06f],[Herbert,06a], [Herbert,06b],[O'Sullivan,06],[O'Flynn,06],[O'Donoghue,05],[Barton,05]

John O'Donoghue
October 2007.

ACKNOWLEDGEMENTS

I would like to acknowledge the genuine support and motivation provided by my supervisor Dr. John Herbert. His continual guidance throughout my research studies in UCC have been of great assistance, I will be forever grateful.

I would like to acknowledge the hardware/software support provided by the Tyndall National Institute under the (SFI) funded National Access Program (NAP). This enabled the design and development of the Tyndall-DMS-Mote which played a major role in the evaluation of the DMS architecture.

Dr. Rossana Salerno-Kennedy, (Department of Medicine, UCC) in helping to identify key medical areas of interest in the development of the DMS.

Stephen Dineen, (Senior Technician, Department of Physiology, UCC) for providing hardware/software assistance and access to the physiology labs during the evaluation of the Tyndall-DMS-Mote.

Mike O' Regan (Information Services Manager, Cork University Hospital) for highlighting real world issues within the medical community.

Chien-Liang Fok (Department of Computer Science, Washington University in St. Louis, USA) in helping to successfully port the Agilla WSN agent platform onto the Tyndall-DMS-Mote.

Rune Fensli (Faculty of Engineering and Science, Agder University College, Norway) in the development of the Data Management System-Validation Model (DMS-VM).

A special thank you to Kai Fei, Yang Han, Gao Ling, Paul Stack and S.J. Min (Taught M.Sc. students, Department of Computer Science, UCC) in the development of various aspects of the DMS architecture during their studies.

I am extremely grateful for the financial support provided by the Department of Computer Science, UCC and the Boole Centre for Research in Informatics (BCRI), UCC.

Finally to my family for their continuous support and encouragement.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACL	Agent Communication Language
API	Application Program Interface
AUML	Agent Unified Modelling Language
BAN	Body Area Network
BPM	Beats Per Minute
CA	Context Awareness
CP	Context Provider
CRKW	Clinical Reasoning Knowledge Warehouse
CRM	Customer Relationship Management
DBI	Desire, Belief, Intention
DMS	Data Management System
DMS-DCM	DMS - Data Consistency Model
DMS-JAI	DMS - Jade Agilla Interface
DMS-UCM	DMS - User Context Model
DMS-UP	DMS - User Profile
DMS-VM	DMS - Validation Model
FIPA	Foundation for Intelligent Physical Agents
FIPA-ACL	FIPA - Agent Communication Language
GSM	Global System for Mobile Communication
IETF	Internet Engineering Task Force
IP	Internet Protocol
JADE	Java Agent Development Environment
JADE-LEAP	JADE-Lightweight Extensible Agent Platform
JADEX	Jade eXtension
JCC	JadeX Control Centre

JESS	Java Expert System Shell
MAS	Multi Agent Systems
OS	Operating System
P2P	Peer to Peer
PC	Personal Computer
PICO	Pervasive Information Community Organization
PLACE	Pervasive Location-Aware Computing Environments
PPC	Patient Point of Care
PDA	Portable Digital Assistant
QoC	Quality of Context
QoD	Quality of Data
QoS	Quality of Service
RFID	Radio Frequency IDentification
RMI	Remote Method Invocation
SMS	Short Messaging System
TCP / IP	Transmission Control Protocol / Internet Protocol
UCM	User Context Model
UML	Unified Modelling Language
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

CHAPTER 1

Introduction

1.1 Introduction

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [Weiser, 91]

The role of pervasive computing brings with it new opportunities to assist us in our daily lives. The term *pervasive* refers to the interconnection of embedded computing devices which can generate large quantities of data. Within a pervasive environment electronic handheld devices are playing an ever increasing role. Smart phones and PDAs now contain sufficient processing and communication resources to enable them to interact with real world objects (sensors, actuators) in real-time [Roussos, 05],[Ballagas, 06]. At present the majority of sensing devices are constrained by their memory, communication and/or power capabilities.

Within a healthcare domain, wireless patient monitoring devices are capable of monitoring specific vital sign datasets. They are able to transmit datasets back to the relevant medical practitioner and/or caregiver's mobile device. [Lorincz, 04],[Drugge, 06],[Lubrin, 05]. By managing these patient datasets in real-time medical errors may be reduced thus providing a higher quality of patient care. At present the majority of medical environments have not fully utilised these pervasive tools.

An intelligent data management infrastructure is needed to realise the potential of these data rich resources. This will enable the mobile users to make well informed decisions and potentially increase their levels of productivity. Presented is the Data Management System (DMS). It is developed with an agent based middleware to intelligently manage data within a pervasive medical environment [O'Donoghue, 05]. The DMS architecture is designed to manage multiple aspects of a pervasive environment ranging from sensor validation to data correlation and context based data delivery. The DMS can deal with the hardware and software properties that may affect the overall quality and delivery of data to its pervasive users.

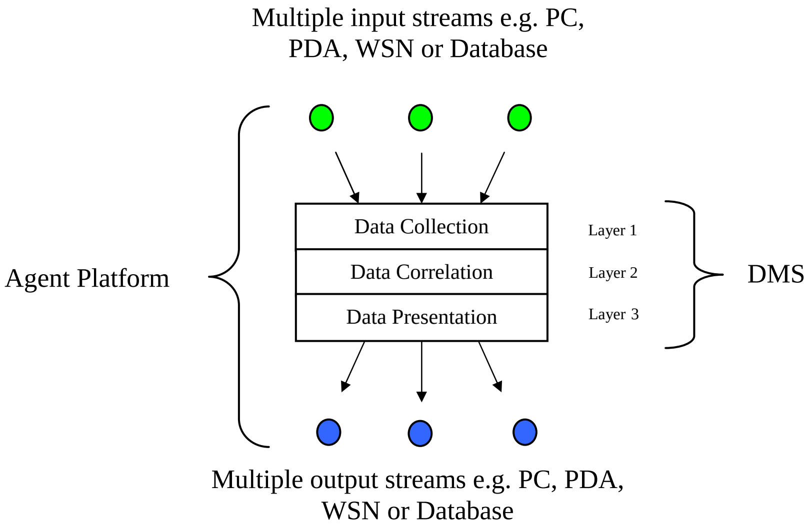


Figure 1.1: DMS Architecture.

The current DMS prototype is designed to manage the complex dynamic datasets within a medical environment. Presented in figure 1.1 is a high level overview of the DMS architecture. It is built on three data management layers:

- Data Collection [Layer 1]

The DMS collects data from wireless patient modules and medical staff PDAs. The frequency of data collection is context based and may be triggered by the ends user's requirement and/or environment context.

- Data Correlation [Layer 2]

Multiple data records relating to the same real world element may exist within the pervasive environment. To ensure that vital information is not overlooked, explicit relationships are created between relevant datasets. This provides a more complete and accurate view of the real world environment.

- Data Presentation [Layer 3]

Finally data presentation is determined by the context of the end user and their personal profile. This will ensure that end users are not overloaded with irrelevant information.

1.2 Application Domain

Pervasive medical environments may produce large quantities of data. Patient datasets need to be collected, correlated and presented in a context aware manner to meet the real-time needs of the mobile medical practitioner. Within this distributed environment a sophisticated middleware infrastructure is needed. It is required to manage all data resources. This work highlights some of the key aspects of a medical based Data Management System (DMS). The DMS aims to effectively manage all datasets through the development of DMS components. The proposed architecture is designed to manage the complex dynamic datasets within a medical environment to improve productivity levels of medical practitioners through the use of software agents. A software agent is ideally suited to function within a context rich environment, due to its built in intelligence and autonomous reasoning. Within a medical environment a high QoS must be maintained at all levels. In relation to data management, one should try to ensure that physicians get the “correct” data on time every time.

Within a medical environment, wireless embedded patient sensing devices have the potential to play a significant role in delivering a higher quality of patient care. They are capable of continuously monitoring a patient’s vital signs and alerting medical practitioners to potential patient risks.

The mobile agent paradigm may play an important role in enabling pervasive computing [Zaslavsky, 04]. Agents are designed to operate effectively within a context rich environment. This dissertation examines the role agents and data management can play within a pervasive medical environment. Issues include data gathering and analysis techniques for multiple data sources (including wireless patient sensing devices, PDAs and smart phones). Data association based on user profiles is also examined in relation to data correlation and dissemination.

1.3 Contributions

The primary contribution of this research is the Data Management System (DMS). It is designed to advance the development of c*ontext-sensitive quality data management for pervasive computing environments*. The DMS consists of a number of agent based DMS components. These components deliver a higher QoS within a pervasive medical based environment through:

- Sensor Validation
- Data Consistency
- Knowledge Based Reasoning
- Context Based Data Delivery
- Distributed Data Management

In collaboration with the Tyndall National Institute and the Department of Medicine (UCC) the Tyndall-DMS-Mote [O'Flynn, 07], [O'Flynn, 06] was developed. It is capable of capturing patient real-time vital sign readings including ECG, pulse, blood pressure (systolic and diastolic) and body temperature. The Tyndall-DMS-Mote enabled real world applications to be developed and played a significant role in the development of the DMS architecture. Issues such as sensor interference and physiological accuracy levels were evaluated to take into account sources of real world QoS degradation.

1.4 Dissertation Structure

This dissertation includes the following chapters:

- Background research and related work within the pervasive community is presented in Chapter 2. Particular areas of interest include the role of pervasive front end mobile devices and the importance of QoS, QoC and QoD within a pervasive environment. Finally the role an effective pervasive middleware can play in "gluing" the pervasive devices together in a homogeneous manner to meet the QoS, QoC and QoD requirements is examined. [O'Donoghue, 05], [Barton, 05], [O'Donoghue, 06f], [O'Flynn, 06], [Angove, 07], [O'Flynn, 07].

- In Chapter 3 an overview of the Data Management System (DMS) architecture and the DMS components is given.

- Pervasive sensing devices are subject to inherent communication and hardware constraints including unreliable wired/wireless network links, interference and limited power reserves. This may result in erroneous datasets being transmitted back to the end user. Presented in Chapter 4 is the DMS-Validation Model (DMS-VM). It validates the sensor readings generated by the Tyndall-DMS-Mote with respect to medically certified sensors. This approach helps to reduce false alarm generation and to identify possible weaknesses within the hardware and software design [O'Donoghue, 06a].

- Data residing on multiple mobile devices and wireless patient motes need to be collected and analysed in a seamless fashion. The Data Consistency component (DMS-DCM) [O'Donoghue, 06b] is presented in Chapter 5. Vital patient datasets may be fragmented over a number of mobile and static devices. If a medical practitioner's mobile device does not contain all known information then the quality of patient care may degrade. Various real world scenarios are analysed in the dissemination and sharing of vital patient datasets within a mobile environment.

- Sensor values originating from a wireless patient mote are typically isolated in nature. This implies that a patient sensor just samples a reading and returns it to a designated device. The Data Management System-Tripartite Ontological Medical Reasoning Model (DMS-TOMRM) [O'Donoghue, 06d] is presented and evaluated in Chapter 6. It is designed to capture sensory data and correlate it with all known data sources (in particular the medical knowledge base and patient profile). This approach reduces the number of "false alarm" generations, thus providing a higher quality of service and patient care.

- The Data Management System-User Context Model (DMS-UCM) component of the DMS architecture is presented in Chapter 7. It highlights the real world relationship between agents and real world context (user schedule and location context). [O'Sullivan, 06]. The DMS-UCM is evaluated for its ability to push context relevant data onto a medical practitioner's mobile device in a timely manner.

- Current mobile and sensing devices contain limited processing and communication capabilities. However, situations do arise where data needs to be processed locally rather than communicated to a central server. Presented in Chapter 8 is the Distributed Data Management Component of the DMS architecture. Here distributed data management scenarios are examined using the Mobile-DMS-Client (Nokia 9500 smart phone) and the Data Management System-Jade Agilla Interface (DMS-JAI).

    - The Mobile-DMS-Client [O'Donoghue, 06c] resides on a smart phone. This architecture is designed to assist patients within their home environment where access to high computation resources is not always available.

    - The DMS-JAI involves the integration of agents on the resource constrained Tyndall-DMS-Mote. The DMS-JAI [Herbert, 06b] utilises the agent capabilities of both middlewares (Agilla agent (mote-based) to Jade agent (PC-based)) to create a single unified agent platform within a pervasive environment.

- In Chapter 9 a conclusion and overall evaluation of the presented DMS architecture is given outlining its contributions. A detailed overview of future work within the field of context-sensitive quality data management for pervasive computing environments and the DMS architecture is given.

CHAPTER 2

Background Research and Related Work

2.1 Introduction

A pervasive environment is made up of numerous interactive mobile devices and embedded sensors. Such infrastructures generate a great deal of data. This may result in data overloading, rendering the benefits of the pervasive applications non-existent.

Within a medical environment, data relating to patients, medical staff and equipment takes on greater importance as it has a direct effect on the delivery of patient care. It is therefore important that relationships between relevant datasets are identified [M. Chagoyen, 04]. The pooling of relevant data sources has the potential to improve patient care. This data correlation helps to provide medical practitioners with relevant real-time information to make a well informed decision.

Within this chapter three key elements in relation to a pervasive environment are examined, they include: 1) Pervasive data, where data originating from patient monitoring and handheld devices are evaluated, 2) Supporting pervasive infrastructures, which provide a platform to help manage and organise the pervasive datasets and 3) A quality based pervasive environment, the factors concerning the delivery of “correct” data to end users. These three elements help to provide a quality orientated medical based pervasive data management infrastructure.

2.2 Pervasive Data

In this section a key medical data source, i.e. wireless patient monitoring device, is assessed along with the handheld devices which utilise real-time and stored datasets to assist medical practitioners.

2.2.1 Wireless Patient Monitoring

At present the vast majority of medical care environments operate on a paper based system. This requires that every relevant patient state is written onto an appropriate form. This approach is not only time consuming but it is prone to human error. In turn a large portion of medical staff's working day is taken up with typing the information into the medical database [Stausberg, 03]. This poor data management approach results in poor patient care while increasing costs.

Pervasive patient care is one approach which may improve or cooperate with the paper based approach [Geer, 06a], [Geer, 06b]. Numerous wireless patient sensing devices have been developed to assist in providing a higher quality of service within the medical environment [Barton, 05] [O'Flynn, 06] [Fensli, 2005] and [Winters, 03]. However data generated by a patient sensing device is not always one hundred percent accurate. Data may be corrupted due to: power/radio interference; poor coding; faulty sensors or poor sensor contact.

In Chapter four of this thesis a single patient vital sign *Beats Per Minute* (BPM) produced by the Tyndall-DMS-Mote is verified by correlating three patient sensors (two ECG and one pulse sensor) simultaneously. This off-line validation process visually displays the Tyndall-DMS-Mote output and helps to identify faulty readings if any. More importantly, these experiments help to improve the medical practitioner's degree of confidence in the Tyndall-DMS-Mote capabilities, in comparison to general purpose patient sensing devices.

2.2.2 Mobile Devices within a Healthcare Environment

The potential role that pervasive computing mobile devices can play within a medical environment is evident [Kearney, 06] [Weinstein, 02], [Banitsas, 04]. A major focus of this thesis has been to identify key areas within a medical environment where data may be gathered, correlated and distributed in an effective manner e.g. DMS-UCM.

The DMS architecture presented in this thesis is targeted towards a mobile healthcare environment. Handheld devices along with environment and patient monitoring sensors are key components in delivering the next generation pervasive healthcare paradigm. A mobile handheld device can enable medical practitioners to access numerous datasets in real-time. This can greatly enhance their levels of productivity. It can also improve patient diagnosis accuracy levels, as the medical data is available at the patient point of care. Handheld medical devices, when integrated correctly (e.g. sufficient data consistency and delivery techniques) can supply the necessary support (data assistance) in providing a higher level of patient care. This can be achieved through the use of pervasive computing resources (e.g. wireless patient sensing nodes) and supporting data management infrastructure. With the emergence of such mobile and sensing devices, issues which may affect their acceptance include: battery power life cycles, radio interference and sensor quality.

2.3 Supporting Pervasive Infrastructures

Distributed computing enables users to share computing capabilities and information stores. A pervasive middleware enables seamless access to remote information resources [Saha, 03]. The infrastructure of a pervasive middleware is made up of a substratum of shared pervasive facilities. It should be interoperable, scalable and QoS enabled [Ding, 06]. Middleware technologies are the key to next generation computing [Sun, 04], [Norman, 98]. A middleware can utilise a networked infrastructure for specific application domains.

A key architectural requirement within a pervasive environment is the attachment of ‘awareness’ to relevant real world objects which we frequently encounter. At present the majority of computing systems are not capable of sensing their environments. They lack the ability to make timely context based decisions. Pervasive computing requires systems and devices with the capability to perceive a real-world context. The majority of current sensing computing systems are reactive in nature. By introducing pro-activeness into our computing systems a number of complications arise including: uncertainty modelling; accuracy in location monitoring and distributed real-time data processing. Another real world factor arises when data from multiple and possibly disagreeing sensors triggers false alarms which may annoy or mislead the user [Saha, 03].

An effective middleware needs to support the run time needs of the end user. Presented in this section is a comparison of traditional distributed middleware with agent based platforms and how they contribute to the pervasive paradigm.

2.3.1 Traditional Distributed Middlewares

Middleware architectures support pervasive context-aware systems by assisting end user applications to deal with the complexity of context specific operations. These may include data acquisition, data reasoning and data dissemination. An effective middleware decouples applications from the underlying heterogeneous sensors, enabling end users to concentrate solely on relevant datasets [Sheikh, 07], [Kavimandan, 06].

Pervasive computing technologies are developed to provide real-time unobtrusive user services in dynamic heterogeneous environments [Shirazi, 04], [Bardram, 04], [Bardram, 07]. General purpose distributed middlewares such as RMI, CORBA or DCOM are traditionally used to provide the necessary communication infrastructure between multiple distributed objects. For a traditional middleware to provide the necessary support within a context rich pervasive environment a complex reasoning layer would need to be developed. This design philosophy forces the development team to use up a great deal of effort in developing the reasoning component (actual link between middleware and front end applications) rather then concentrating on the main context architecture. With software agents the context reasoning is built into the core middleware philosophy. For example, the Jade agent middleware is built over RMI. Jade is sufficiently decoupled from the RMI run time requirements, enabling the developer to concentrate on the key context handling procedures.

2.3.2 Agent Based Middlewares

Agent technology is the enabling middleware utilised by the distributed components within the DMS presented in this thesis. In the context of software engineering, an agent can be defined as:

"An entity within a computer system environment that is capable of flexible, autonomous actions with the aim of complying with its design objectives" *[Wooldridge, 97]*

A pervasive environment is complex in nature; it requires sophisticated communication and reasoning infrastructures to meet the end user's run time requirements. An agent-based middleware contains sufficient autonomy, reactivity, pro-activeness and social ability to facilitate the development of context based applications [Wooldridge, 97], [Zaslavsky, 04], [Bisdikian, 02]. The field of agent technology is seen as a highly suitable paradigm and communication infrastructure for the analysis and design of mobile healthcare systems [Della Mea, 01]. An agent middleware manages the coordination, cooperation and interoperability of distributed components by linking applications with their underlying low level software and hardware infrastructure. Middleware infrastructures operate within a dynamic environment. A balance is needed between transparency and context awareness, to ensure it does not intrude on the end user's daily activities [Soldatos, 07]. The majority of agent based middlewares are designed to communicate, migrate and have a certain degree of built in autonomy. This enables them to trigger events with little or no intrusion on the end user (specifically routine events). Agent middlewares can be extremely effective when combined with specialised hardware or software elements. To fully understand the context of any environment, sensory data needs to be obtained and adequately processed (i.e. cognitively understood). By merging an intelligent middleware with specialised sensory data analysis tools, relevant context datasets may be collected, correlated and disseminated to interested parties within the network. As middlewares in general operate at a very high logical level, the use of lower level applications/tools maximises the effectiveness of both domains in delivering on the user's real-time requirements.

Software agents exhibit a number of social characteristics, desires, beliefs and intentions (DBI) [Rao, 91]. The use of agents within a healthcare environment has been shown to increase productivity in assisting medical staff during their daily tasks. [Mazzi, 01] outlined the concept that, "*Software agents can provide* *an extension of the doctor by interacting with the patient via a computer*". The role of any medical practitioner is multifunctional. Software agents are capable of reading in multiple context elements which may assist the medical practitioner. For example:

- Scenario 1 [Timing and Mobility]

Andrew, a new patient at his local hospital, had been complaining of a slight chest pain. After initial tests doctors could not find anything in particular but agreed to keep him under observation for the next few days. Wireless monitoring sensors were attached to Andrew. This enabled him to walk around the hospital grounds while doctors were able to monitor his condition on a constant basis. A few hours later the wireless sensors indicated that Andrew's heart rate had reached a serious level and that he needed instant attention. Software agents may be configured to react as follows:

  - Instantly notify relevant practitioners of Andrew's state and location.
  - Selected medical staff PDAs may be instantly uploaded with Andrew's real-time and archived data.

- Scenario 2 [Informed Decision]

Michelle, the first resident doctor to attend Andrew, could see that Andrew was just after suffering a heart attack. After examining Andrews's condition she decided to administer a thrombolytic drug. Before carrying out the procedure the software agents reported that Andrew was allergic to such drugs and identified another course of action.

[Mazzi, 01] presented a number of scenarios where software agents may be broken down into logical data analysing tools to assist the medical staff in providing a continuous effective service. Three basic agent types were identified: 1) Personal Assistant Agent (an interface between the user and the computing environment. Here user requests are interpreted and relevant agents are called upon to execute the tasks), 2) Search Agent (searches and retrieves information based on the user's request.) and 3) Patient-Monitoring Agent (manage the data generated by the wireless monitoring device).

The design philosophy inherently built into a software agent 1) helps the development teams to focus on key end user's context requirements and 2) provides the necessary middleware support infrastructure to manage real world events as outlined in scenarios 1 and 2 above.

2.4 A Quality Based Pervasive Environment

Smart mobile devices, embedded sensors and high bandwidth networking infrastructures are assisting the pervasive paradigm. A sophisticated middleware and associated infrastructure provide the necessary tools to enable a number of quality oriented data aspects within the pervasive domain. These include Quality of Service (QoS), Quality of Context (QoC) and Quality of Data (QoD). The term 'QoS' is typically associated with the performance parameters required for communication/power protocols or other temporal run-time requirements. The term 'QoC' generally implies the delivery of relevant data at the correct moment to the end user. QoD is the delivery of the "correct" data in association with the QoS and QoC user run-time requirements.

2.4.1 Quality of Service

Aside from the fundamental infrastructures required to transmit and store relevant real-time information a number of QoS components must be integrated. This will ensure that real-world issues such as intermittent communication failures or faulty sensing devices are recognised. Such information can then be passed on to the end user where a final acceptance or rejection of the sent data can be processed [Varshney, 06], [Khoukhi, 05]

Wireless networks and interacting devices are key elements within a pervasive environment. Wireless networks are prone to higher bit error rate and interference. This may result in a degraded service and potential bottlenecks [Choi, 04]. To reduce the inherent service issues, QoS protocols are designed to monitor key network parameters to identify potential network failure points. They are also designed to provide alternative solutions to intermittent network connections at run-time.

Multiple mobile devices require a common communication infrastructure if they are to operate in a seamless manner. [Shirazi, 04] presents a PICO (Pervasive Information Community Organization) middleware. The PICO middleware provides a specific higher QoS through the deployment of a *just-in-time* approach where distributed middleware elements interact with the mobile device or server backend applications. This technique improves efficiency by providing the services when required, thus saving on valuable resources.

2.4.2 Quality of Context

"Quality of Context (QoC) is any inherent information that describes context information and can be used to determine the worth of the information for a specific application. This includes information about the provisioning process the information has undergone (e.g. history, age)" [Krause, 05]

A distributed middleware provides the necessary services which enable front end and backend devices/applications to interact as a single entity. These may include DCOM, CORBA, RMI or agent based middlewares Aglets, Jade and Agilla. [Kisazumi, 03] presents CAMPUS a context-aware middleware to construct context-aware applications. The CAMPUS data gathering approach shares similar qualities to the DMS architecture. Data is pooled from sensors, network PCs and distributed databases. Context events are managed by Jade or JadeX agents. They may also be filtered through the use of an external expert systems (e.g. Jess). Both CAMPUS and DMS continuously monitor the QoC parameters to ensure that data passed on to the mobile user is relevant to the user's task at hand.

An effective QoC based application provides the end user/application with relevant real-time information in relation to their real-world environment [Buchholz, 03]. [Huebscher, 04] presents an adaptive middleware for context aware applications. Here the location of the mobile user is matched against all known real-world devices which are of interest.

2.4.3 Quality of Data

A single data element within a pervasive environment may be viewed from many perspectives including completeness, freshness, trustworthiness and context.

Metadata enables software developers to describe data and its various parameters. [Mihaila, 00] outlined a QoD metadata for data source selection and ranking. In developing a metadata model parameters may now be integrated to improve data selection.

Within the pervasive computing domain QoS and QoC are well defined, QoD is not. In relation to this thesis the term QoD is defined as providing the end user with value added data which enables them to fully understand where the data originated from and how the final value was derived. It also helps to ensure that the end user receives the correct data on time every time.

2.5 Summary

A typical pervasive environment is made up of a large number of interacting sensing and processing devices. It is imperative that such devices are able to communicate and interact in real-time to meet the end user's run time requirements. To provide a high quality service within such an environment all QoS, QoC and QoD parameters need to be satisfied.

A large number of distributed middlewares exist. Each has its own strengths and weaknesses. Within a pervasive environment numerous events are generated by users and system devices. A reaction to each event should be tailored to meet the end user's run time and data quality requirements. An agent middleware is used for its ability to intelligently filter non-relevant real world events and to seamlessly interact with end users in a non-intrusive manner. The agent platform also provides a mature computing paradigm. It contains an inherent sentient design philosophy which directs the software developer to focus on key user context and data quality requirements.

CHAPTER 3

Data Management System Architecture

Pervasive environments generate large quantities of data. If utilised correctly these datasets may assist the end user to conduct their daily tasks in an effective manner. Otherwise the data gathered may become redundant. It is important that data management infrastructures are developed to interact with the end user in a context and situation aware manner. A new data management system for context based pervasive environments named DMS is proposed [O'Donoghue, 05], [O'Donoghue, 06f]. The DMS is designed to gather all known data (static and dynamic) and provide the end user with relevant context based datasets. This has the potential to increase the end user's level of productivity and reduce information overload.

3.1 Pervasive Data Management Architecture

Presented in this thesis is the Data Management System (cf. figure 1.1). It is built on an agent middleware and supports context based data management within a pervasive environment. Outlined in figure 3.1 is an overview of the context based pervasive data management architecture. A pervasive environment consists of 1) Front end pervasive elements (PDAs, patient motes, smart phones) that have limited communication and processing capabilities and 2) Backend servers, where processing and communication are less of an issue (cf. figure 3.1). The backend sources provide the front end devices with context based or user requested datasets.

Data needs to be intelligently distributed to ensure that it is delivered to the end user in a timely manner. To assist in achieving this goal sufficient reasoning facilities must reside within the front end devices and backend servers. This provides context based communication, enabling data to be transmitted in a coordinated and timely manner. An agent middleware (Jade) is utilised to provide the necessary communication and reasoning capabilities between the front end and backend devices.

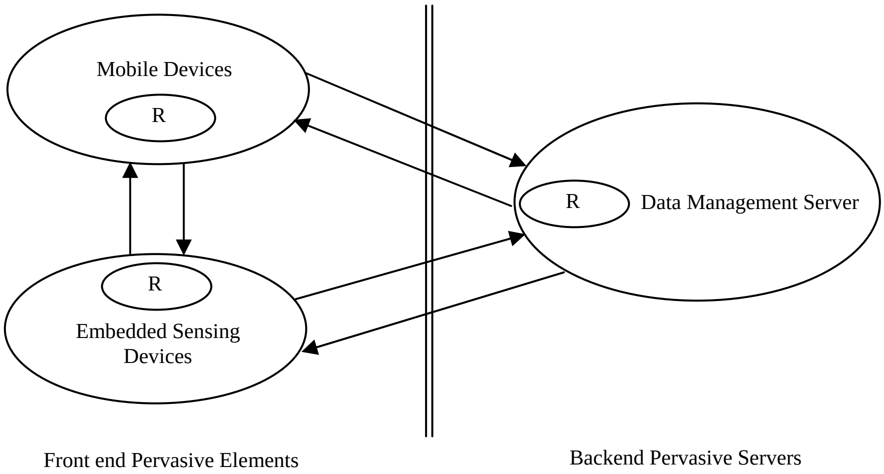


Figure 3.1: Context Based Pervasive Data Management.

Illustrated in figure 1.1 is an overview of the DMS architecture where R represents an inbuilt reasoning capability. It contains three fundamental data management layers designed to deliver an effective context based service. These are data collection, data correlation and data presentation. The functionality of the DMS layers is provided through the development of data management components. DMS Components are designed to function at various levels within the DMS from low level data gathering to high level reasoning. A DMS component provides specific data management services. Five DMS components are created and provide the primary functionality of the DMS architecture (cf. figure 3.2). The DMS presented in this thesis is healthcare oriented but the components developed may be carried over to other application domains, such as banking, factory automation or CRM (Customer Relationship Management) based models.

1) Sensor Validation Component

A sensor may generate large quantities of data over a period of time. It is possible for it to become unstable and produce inaccurate data. This in turn may trigger incorrect system events resulting in partial or complete system failures. It is vital that real-time datasets, particularly of a life critical nature, are validated. This helps to increase the level of trust for the data generated by the pervasive sensors.

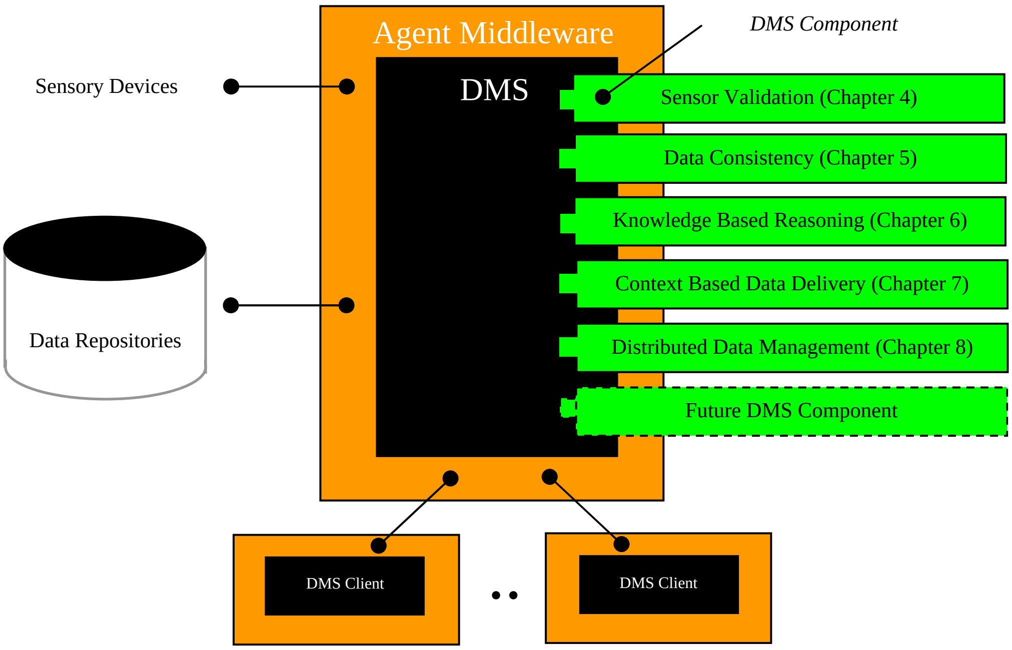


Figure 3.2: A Logical Overview of the DMS Architecture and DMS Components. Mobile DMS Clients may interact with the DMS Server. Data Pools Include Sensory Devices and Large Data Storage Facilities.

2) Data Consistency Component

Data resides in a large range of devices and may change over a period of time. Within a mobile environment, data should be seamlessly available to the end user. The Data Consistency component ensures that data on a mobile device (front end) is synchronised with all known data sources (front end and backend).

3) Knowledge Based Reasoning Component

An isolated data element within a pervasive environment may only be recognised as a single contextual fact. To fully utilise these contextual facts it is necessary to merge all known related datasets to provide a complete context overview for the end user. Partial contextual overviews may generate incorrect system events. The knowledge based reasoning component of the DMS architecture pools all available information to provide the end user with a contextual (complete or incomplete) overview of their current real world state. This provides a contextual analysis of known data elements within the data correlation and presentation layers.

4) Context Based Data Delivery Component

The real-time identification of an end user's location may play an important role in delivering an effective context based data management service. One possible technique in delivering relevant context specific datasets to the end user is through the merger of an end user's profile (user ID, daily schedule) with their real-time location. This approach helps to reduce information overload and provide the end user with relevant real-time information. The context based data delivery component provides an essential service within the data correlation and presentation layers within the DMS architecture.

5) Distributed Data Management Component

Situations do arise where data needs to be processed locally. The Distributed Data Management Component of the DMS architect evaluates two key areas 1) Processing through the utilisation of smart devices (e.g. smart mobile phone) to manage local patient sensory data and 2) WSN agent integrating with resource rich mobile devices. Agent middlewares are utilised to manage context based data. DMS agents operating within a WSN are designed to communicate with agents functioning within resource rich devices. An evaluation of the overhead created through the deployment of agent middleware on resource constrained nodes is given.

3.2 Summary

Each of the three DMS layers may collaborate or function in isolation. The design of a DMS component is application specific. Therefore if greater attention is required for sensor analysis then suitable procedures or methods may be integrated at the data collection layer. The five DMS component prototypes presented in this thesis are designed to target real world data management medical scenarios.

CHAPTER 4

Sensor Validation within a Pervasive Medical Environment

4.1 Introduction

Pervasive patient sensing devices may generate large quantities of data. This data needs to be transmitted to central medical servers or mobile devices for real-time analysis. Various factors can affect the ‘quality’ of our patient data. These include: wireless interference (e.g. access point or radio failure) and/or sensor failure. Vital patient data packets may be lost resulting in an incorrect diagnosis. Patient sensor failure is a reality. It is imperative that sensor failure is detected within a set of real time boundaries to ensure a higher QoS is provided. Presented is a Data Management System-Validation Model (DMS-VM) [O’Donoghue, 06a], the sensor validation component within the DMS architecture. It is designed to manage sensor failure and interference in a controlled and intelligent manner. The DMS-VM is capable of sampling multiple patient vital sign readings simultaneously. It is then able to intelligently analyse the sensory datasets to verify their integrity. This novel approach provides a higher QoS within a context aware medical environment.

One of the core design features of the DMS (Data Management System) [O’Donoghue, 06f] is to provide a higher QoS within a pervasive environment. The DMS-VM includes two validation protocols to ensure medical practitioners receive accurate patient datasets. The first protocol is “isolated sensor validation” (e.g. where a single sensor is sampled and compared against a set of predefined ranges). The second protocol is “cross sensor validation” (e.g. where two or more patient sensor types are compared against each other (for example an ECG sensor R-R interval (i.e. time interval between heart beats, cf. figure 4.4a) against a pulse sensor). The loss of transmitted signals within the medical community is unacceptable. [Golmie, 05] demonstrated that transmitted ECG signals from WPANs (Wireless Personal Area Network) suffered poor reliability due to radio interference. Sensor failures are an additional concern which may result in inaccurate datasets.

4.2 DMS-VM

A logical overview of the DMS-VM architecture is presented (cf. figure 4.1) [O'Donoghue, 06a]. If a patient monitoring device does not contain sufficient processing capabilities, data is transmitted to the mobile device or central server. "Isolated sensor validation" may be executed within the mobile device (e.g. PDA, mobile phone) or on the DMS-Server. "Cross sensor validation" is executed within the DMS-Server. Here, datasets are collected and compared against one another to verify their integrity. This will enable the DMS-VM to determine if the R-R intervals (i.e. period of time between pulse signals) are producing similar results.

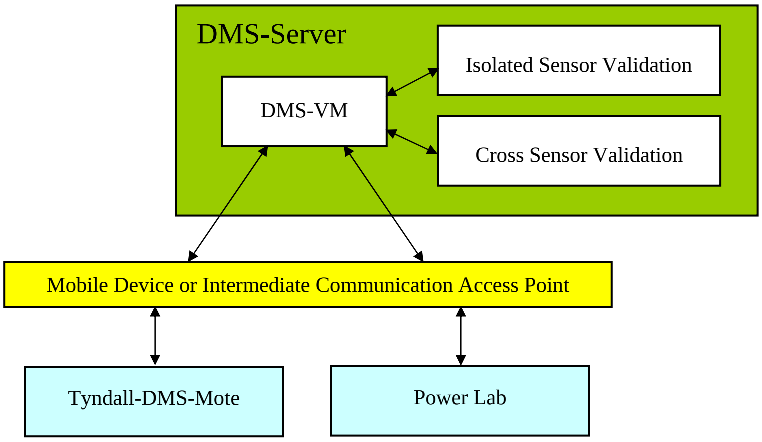


Figure 4.1: The DMS-VM Architecture. Power Lab is an ADInstruments system used to capture and analyse biometric signals [ADInstruments, 07].

An outline of some of the key software agents within the DMS-VM is illustrated in figure 4.2. These include:

- **Mobile Device Manager Agent (MDMA)**

The resident JADE software agent is designed to arbitrate between the central DMS-Server and medical practitioner. This agent deals with medical staff requests and incoming server updates. The MDMA interacts with the Mobile Device Data Validation Agent for pre and post validation checks.

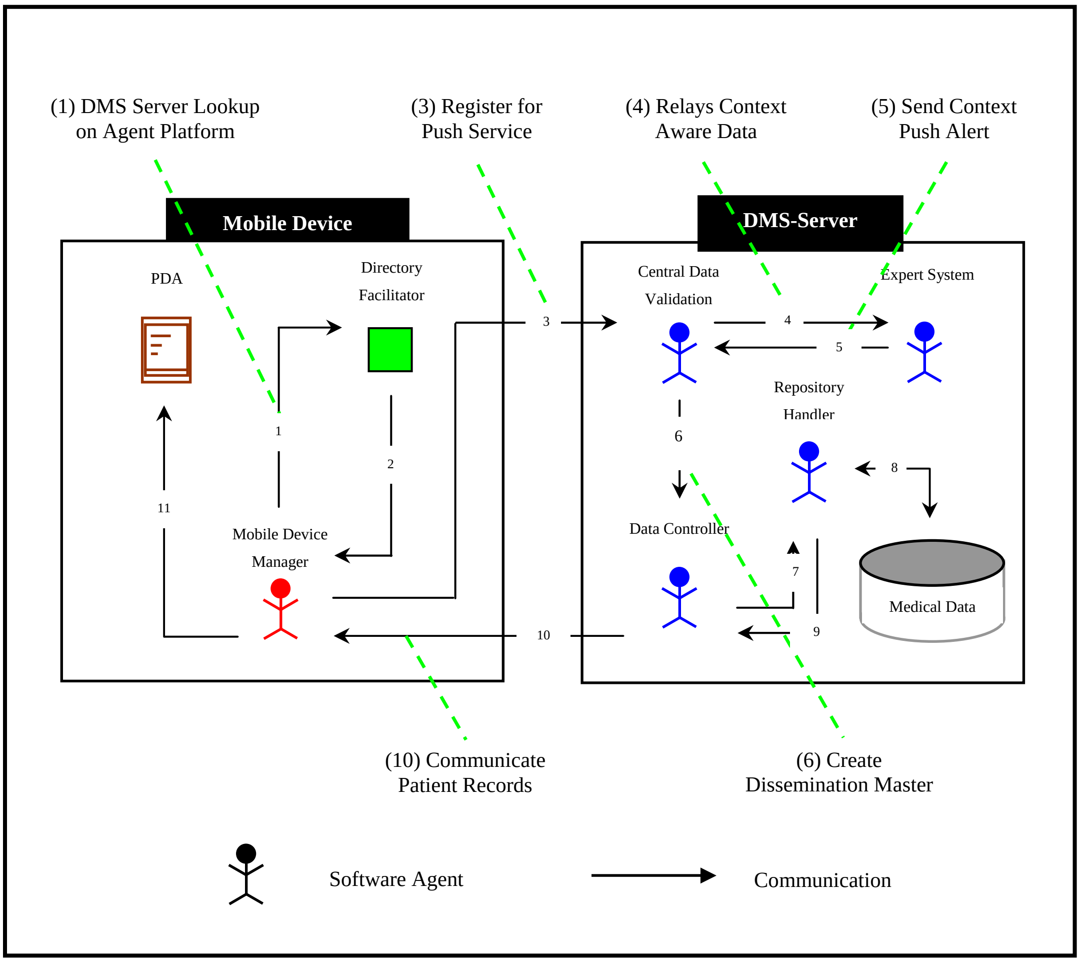


Figure 4.2: The DMS-VM Agent Based Architecture.

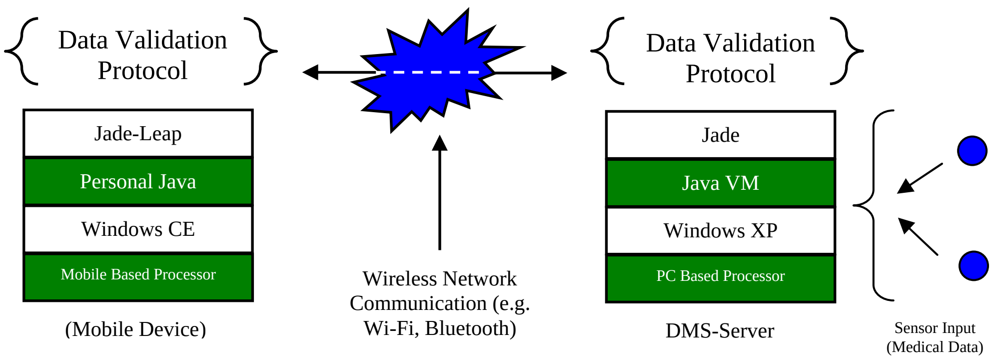


Figure 4.3: The DMS-VM Implementation Overview.

- **Mobile Device Data Validation Agent (MDDVA)**

Incoming and outgoing datasets to and from the mobile device pass through the MDDVA. This agent is responsible for ensuring that the current dataset resident on the mobile device is within a set of predefined sensor ranges. It achieves this by associating each sensor type with a collection of sensor type parameters.

- **Central Data Validation Agent (CDVA)**

The CDVA manages all real-time sensor data streams and medical practitioner request/updates. All of the CDVA executions are based on the central expert system, which contains a formal set of data validation rules.

- **Expert System Agent (ESA)**

The ESA associates incoming sensor types with a set of predefined validation rules.

- **Data Controller Agent (DCA)**

The DCA populates the medical data store with incoming medical practitioner request/updates and sensor values.

The Directory Facilitator (DF) is a JADE multi-agent management facility. It maintains a continuous view of all JADE agents within its local platform. This enables mobile devices and backend servers on the same platform to communicate. The DF continuously examines the services each mobile agent provides, their location (i.e. device) and current state. If a JADE agent is created or terminated local platform agents will be notified.

The DMS-VM prototype enables single or multiple sensor readings to be validated. It is built on a software agent middleware JADE [Bellifemine, 99] (cf. figure 4.3). JADE is a software agent middleware that resides on the DMS-Server. JADE contains sufficient reasoning and autonomous capabilities. It was selected as the underlining middleware to meet the DMS data management requirements. The JADE middleware was selected based in it ability to easily integrate with all DMS components which are primarily Java based. Jade-Leap [Berger, 03] operates on mobile devices (a medical practitioner’s PDA or smart mobile phone).

The DMS-VM is capable of simultaneously comparing information sampled from different sensors. This will increase confidence and decrease the number of false alarm generations as erroneous sensor readings can be filtered out.

4.2.1 Evaluation

The DMS-VM protocols “Isolated Sensor Validation” and “Cross Sensor Validation” were implemented to evaluate the reliability of three physiological vital sign patient sensors which consisted of 1) Power Lab Pulse, 2) Power Lab ECG and 3) Tyndall-DMS-Mote ECG sensors. The Power Lab system is a commercial product [ADInstruments, 07] and contains a number of fully certified medical patient sensing devices. The Tyndall-DMS-Mote is a wearable mobile patient monitoring device designed to sample ECG, blood pressure (systolic and diastolic), pulse and body temperature readings. It was developed by the Tyndall National Institute [Tyndall, 07].

The key comparison of all three sensors is cardiovascular measurement “Beats Per Minute” (BPM). An evaluation of the DMS-VM protocols over three physiology sensors is conducted. This is achieved by determining the BPM from the Power Lab ECG, Power Lab Pulse and the Tyndall-DMS-Mote ECG sensory datasets. Two physiological analysis techniques were created (Area and Slope technique) to determine the BPM for each sensory dataset.

4.2.2 Analysing Physiological Signals

Presented in Figures 4.4 (A), (B) and (C) are the Power Lab Pulse, Power Lab ECG and Tyndall-DMS-Mote ECG physiological signals respectively. These readings were sampled simultaneously while the patient was in a resting state. They represent the most accurate readings achieved with the hardware during DMS-VM evaluation.

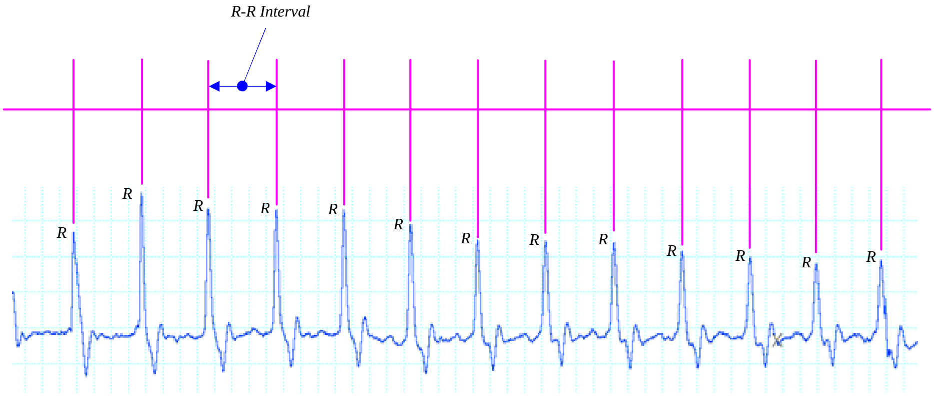


Figure 4.4 (A): Power Lab Pulse Signal. Note the drop off in the height of each pulse peak. This is a natural phenomenon as the patient’s heart begins to relax thus producing less force resulting in lower R points or pulse peaks. A single R-R interval is equivalent to one pulse cycle.

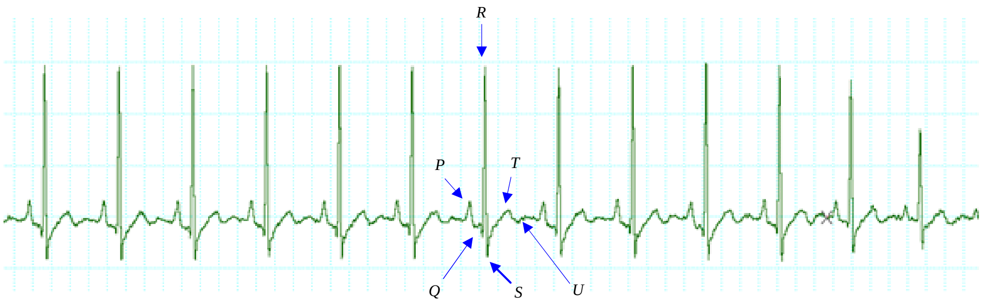


Figure 4.4 (B): Power Lab ECG Signal. Note P, Q, R, S, T and U Points are well defined.

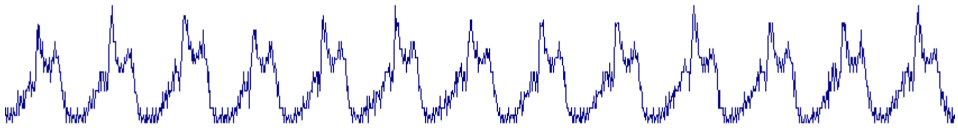


Figure 4.4 (C): Tyndall-DMS-Mote ECG Signal. Note P, Q, S, T and U points are not identifiable due to power interference. For the purposes of the experiments presented in this thesis only the pulse peaks or R-R intervals of all three sensor types are utilised. This enables the BPM to be reliability calculated over a period of time.

4.2.3 Averaging Technique

ECG signals are extremely sensitive as they are required to measure small amounts of voltage change generated by the human heart typically in mVs (millivolts). ECG signals are susceptible to external magnetic and electronic fields resulting in poor signal quality. Presented in figure 4.5 (A) and (B) are examples of interference which need to be taken into account during extraction of specific datasets (for example R-R interval).

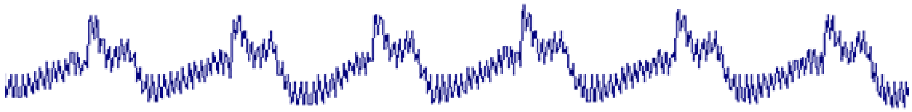


Figure 4.5 (A): Poor Quality Tyndall-DMS-Mote ECG Signal due to power interference.

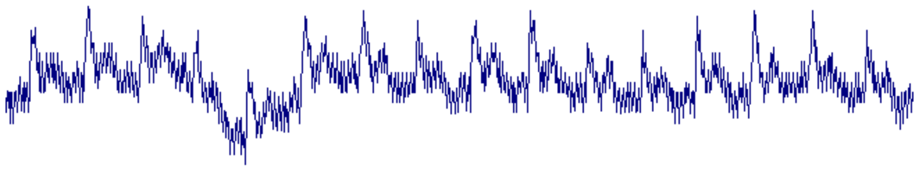


Figure 4.5 (B): Poor Quality Tyndall-DMS-Mote ECG Signal Due to Power Interference and Baseline Wandering.

Analysing a patient signal is a complex task. Signals frequently contain levels of interference which may result in the generation of incorrect datasets. Ideally physiological signals need to be well defined and not contain interference. Presented in figure 4.6 (A), (B), (C) and (D) are the various stages of how a Tyndall-DMS-Mote ECG signal may be averaged (filtered) to remove minuscule interference points.

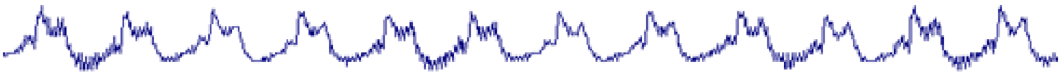


Figure 4.6 (A): Raw Tyndall-DMS-Mote ECG Signal with Twelve Heart Beats and Continuous Interference Points.

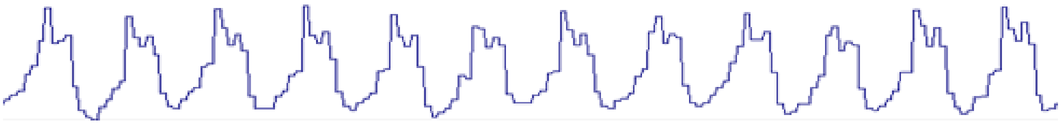


Figure 4.6 (B): A Refined Tyndall-DMS-Mote ECG signal after averaging five points from the original signal with a count of twelve heart beats.

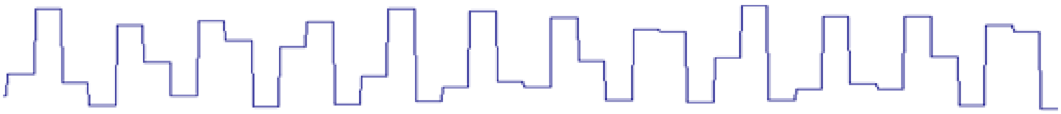


Figure 4.6 (C): A Refined Tyndall-DMS-Mote ECG signal after averaging ten points from the original signal with a count of twelve heart beats.

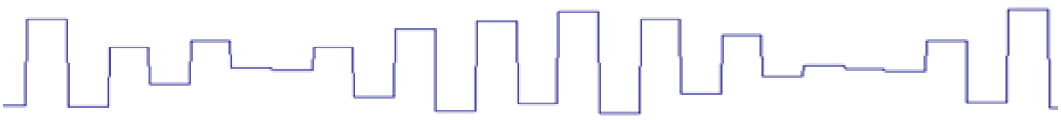


Figure 4.6 (D): An Over Refined Tyndall-DMS-Mote ECG Signal after Averaging 15 Points from the Original with Eleven Heart Beats. Averaging can remove Fuzziness from the Original Noisy Signal. Over Averaging a Signal may Result in a Loss of Data i.e. Heart Beats 10 and 11 may be seen as One Cycle resulting in a Count of 11 Heart Beats.

As the human heart beat increases the R-R intervals decrease and conversely as the human heart beat decreases the R-R intervals increase. This natural fluctuation in the human heart may be utilised to dynamically select an appropriate averaging rate thus providing greater accuracy.

4.2.4 Area Technique

The Area technique is created to analyse a Pulse or ECG signal to identify the BPM over a period of time. Presented in figure 4.7 (A) is a Tyndall-DMS-Mote ECG signal. A 'base bar' is manually factored in prior to code execution. The area of interest is represented as triangles (cf. figure 4.7 (A)) the base of the triangle is the user specified 'base bar'. These triangles represent the sample area. A sample area signifies the minimal allowable area during the identification of a valid cycle. An ECG or Pulse signal which produces an area (above the base bar) and less then the sample area is recorded as an invalid heart beat (cf. figure 4.7 (B)).

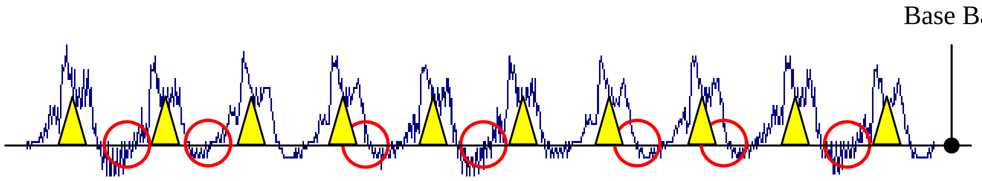


Figure 4.7 (A): A Tyndall-DMS-Mote ECG Signal with Undefined Points of Intersection between the ECG Signal and Base Bar.

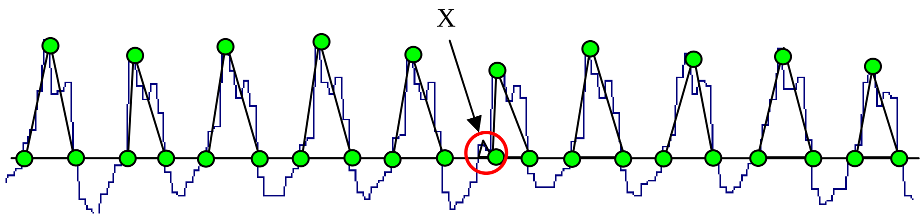


Figure 4.7 (B): A Tyndall-DMS-Mote ECG Signal (Averaging Technique of 5). Points of Intersection are well defined. An Area which is less then the Sample Area is Regarded as an Invalid Heart Beat (point x).

Two Area techniques were created Static and Dynamic. The Static technique operates with a fixed "sample area". The 'base bar' is evaluated in three positions. They are 25%, 50% and 75% from the average peak of the last recorded ten cycles. A valid cycle is viewed when the minimum and maximum sample areas fit in between the base bar and the average peak. The Dynamic technique alters the level of the "base bar" and calculates a new "sample area" based on previous recorded cycles.

4.2.5 Slope Technique

The slope technique is created to analyse a Pulse or ECG signal to identify the BPM over a period of time. A patient's physiological signal (graph) is made up of a large numbers of points. Each point has an X, Y coordinate. X represents time and Y represents height (strength of heart beat). Over a period of time the physiological sensor values provide a graph which represents the pulse or heart beat. Presented in figure 4.8 (A) is a Tyndall-DMS-Mote ECG signal (with an average technique of 10).

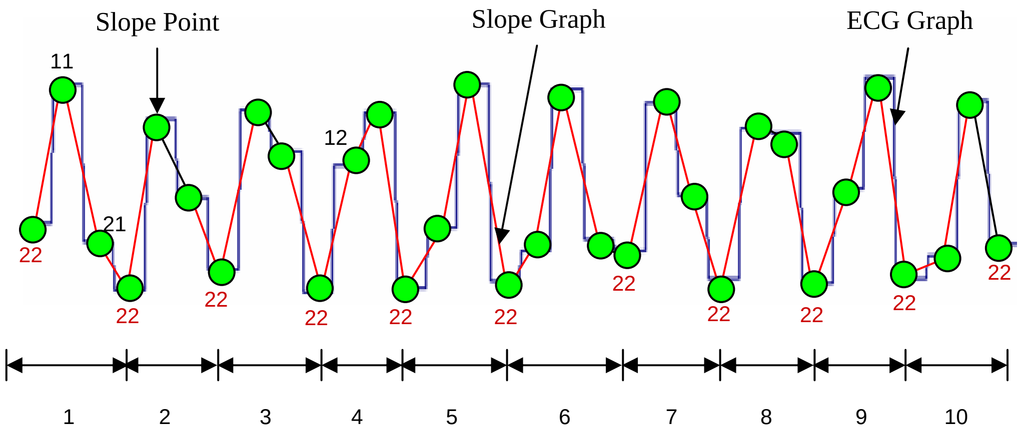


Figure 4.8 (A): Refined Tyndall-DMS-Mote ECG Signal (Averaging Technique of 10). A change in the direction of the graph (i.e. an up or down direction) is represented as a slope point (cf. figure 4.8 (B)). Once identified the slope points make up a slope graph. This provides a well defined graph to calculate the BPM.

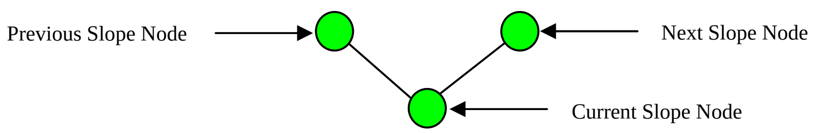


Figure 4.8 (B): Slope Identification Technique. The Direction (i.e. up or down) of the Previous and Next Nodes, Identify the Slope Identification of the Current Node. A Number "22" slope Node is a Base Node (when the slope graph changes from a downward to an upward slope). The X Distance between the Number "22" Nodes is measured in Fractions of a Second and Represent the Length of a Single Cycle (heart beat).

4.3 Test Case Environment

An evaluation of the Static and Dynamic Area and Slope techniques are given. An averaging technique of 10 was applied to the original noisy signals. This provided a clean graph to conduct the experiments. Two interference experiments were also conducted 1) Power interference (A mains power source was placed near the Power lab and Tyndall sensors) 2) Hand Tap interference (The Pulse and ECG sensors were subject to physical contact during recording). Sample C code is given in Appendix A.

All BPM numbers are compared against the BPM count of the Power Lab Pulse as it is a medically certified device and constantly achieved 99% accuracy under all conditions. This is referred to as the *golden standard*.

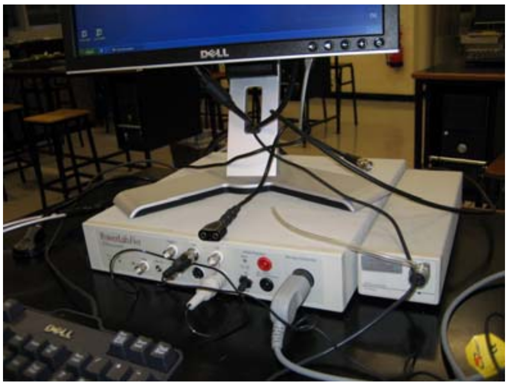


Figure 4.9: (A) The Power Lab 4/25T is an integrated data recording unit. Here a Power Lab Pulse and ECG sensors are attached.

Figure 4.9: (B) The Tyndall-DMS-Mote with three ECG cables attached, positive, negative and ground.

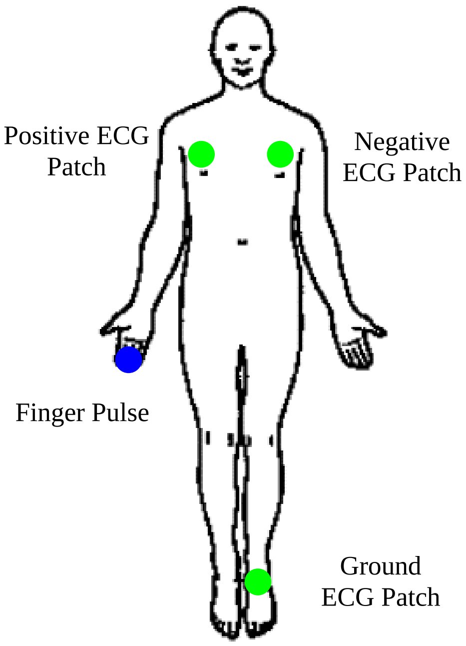


Figure 4.9: (C) Placement of ECG Patches and Pulse sensor.

Figure 4.9: (D) The Power Lab and Tyndall sensors attached during data gathering experiments.

4.4 Evaluation Results

Results are gathered to evaluate the Area and Slope techniques. These techniques are applied to Power Lab Pulse, ECG and Tyndall-DMS-Mote ECG signals which are sampled under various hardware and patient conditions. Hardware items and testing environments are presented in figure 4.9 (A), (B), (C) and (D). Detailed results are given in Appendix B and D. The processing time required to execute each algorithm was negligible and analysis of algorithm performance is outside the scope of this thesis.

4.4.1 An Overview of the Area and Slope Techniques

No.		BPM
	Power Lab Pulse, Visual	68
1	Power Lab Pulse, Static Area (25%)	0
2	Power Lab Pulse, Static Area (50%)	11
3	Power Lab Pulse, Static Area (75%)	65
4	Power Lab Pulse, Dynamic Area (10)	32
5	Power Lab Pulse, Dynamic Area (20)	37
6	Power Lab Pulse, Dynamic Area (30)	41
7	Power Lab Pulse, Static Slope (5)	68
8	Power Lab Pulse, Static Slope (10)	52
9	Power Lab Pulse, Static Slope (20)	18
10	Power Lab Pulse, Dynamic Slope (20)	68
11	Power Lab Pulse, Dynamic Slope (40)	68
12	Power Lab Pulse, Dynamic Slope (60)	68

Table 4.1: Complete List of Analysis Techniques applied to the Power Lab Pulse Sensor with a patient state of resting and Power Interference.

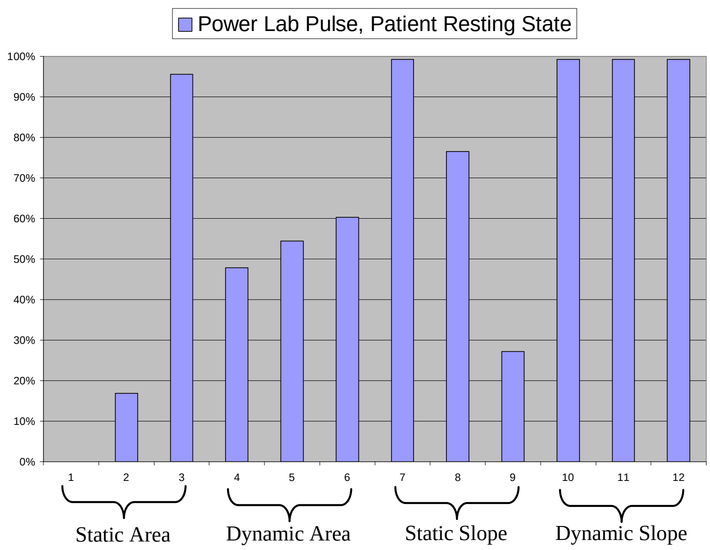


Figure 4.10 (A): An overview of the Area and Slope Techniques Applied to the Power Lab Pulse Sensor with a patient state of resting and Power Interference (cf. Table 4.1).

An overview of the Static and Dynamic Area and Slope techniques are presented in table 4.1 and figure 4.10 (A) as applied to the Power Lab Pulse. These readings were taken with the patient in a resting state. The Area techniques produced an average of 46% in relation to the golden standard. Calculating the specific area of a noisy graph is complex in nature and requires a great deal of analysis, not only of the current cycle under review but the recorded cycles. The Static and Dynamic Slope techniques are based on a simple principle of following the general direction of the graph (cf. figure 4.8 (B)).

No.		BPM
	Power Lab ECG, Visual	68
1	Power Lab ECG, Static Area (25%)	0
2	Power Lab ECG, Static Area (50%)	19
3	Power Lab ECG, Static Area (75%)	68
4	Power Lab ECG, Dynamic Area (10)	37
5	Power Lab ECG, Dynamic Area (20)	36
6	Power Lab ECG, Dynamic Area (30)	40
7	Power Lab ECG, Static Slope (5)	68
8	Power Lab ECG, Static Slope (10)	52
9	Power Lab ECG, Static Slope (20)	18
10	Power Lab ECG, Dynamic Slope (20)	68
11	Power Lab ECG, Dynamic Slope (40)	68
12	Power Lab ECG, Dynamic Slope (60)	68

Table 4.2: Complete List of Analysis Techniques applied to the Power Lab ECG Sensor with Power Interference.

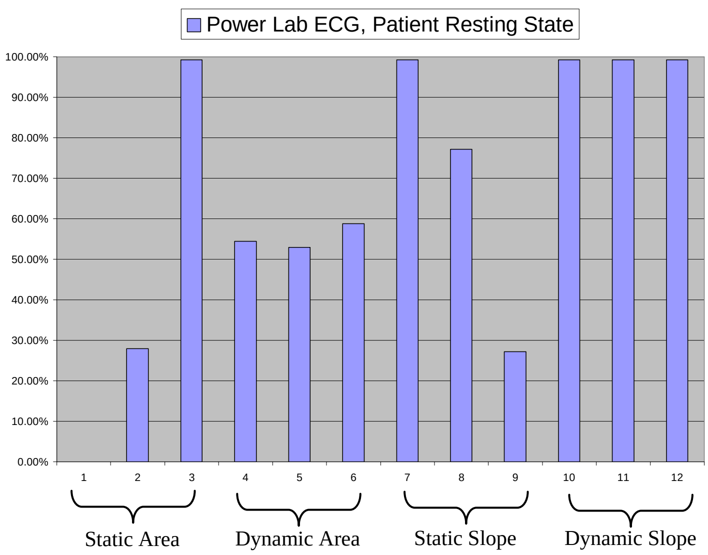


Figure 4.10 (B): An overview of the Area and Dynamic Techniques Applied to the Power Lab ECG Sensor with Power Interference (cf. Table 4.2).

The Power Lab ECG with a patient resting state produced similar results (cf. figure 4.10 (B)) as the Power Lab Pulse. The Dynamic Slope technique is proving to be extremely consistent under all conditions.

No.		BPM
	Tyndall-DMS-Mote ECG, Visual	59
1	Tyndall-DMS-Mote ECG, Static Area (25%)	108
2	Tyndall-DMS-Mote ECG, Static Area (50%)	75
3	Tyndall-DMS-Mote ECG, Static Area (75%)	98
4	Tyndall-DMS-Mote ECG, Dynamic Area (10)	55
5	Tyndall-DMS-Mote ECG, Dynamic Area (20)	58
6	Tyndall-DMS-Mote ECG, Dynamic Area (30)	56
7	Tyndall-DMS-Mote ECG, Static Slope (5)	***184***
8	Tyndall-DMS-Mote ECG, Static Slope (10)	57
9	Tyndall-DMS-Mote ECG, Static Slope (20)	50
10	Tyndall-DMS-Mote ECG, Dynamic Slope (20)	68
11	Tyndall-DMS-Mote ECG, Dynamic Slope (40)	58
12	Tyndall-DMS-Mote ECG, Dynamic Slope (60)	60

Table 4.3: Complete List of Analysis Techniques applied to the Tyndall-DMS-Mote ECG Sensor. Static Slope 5 produces an erroneous number due to the noise in the ECG signal (power interference generated a large number of up and down direction nodes (cf. figure 4.8 (B)).

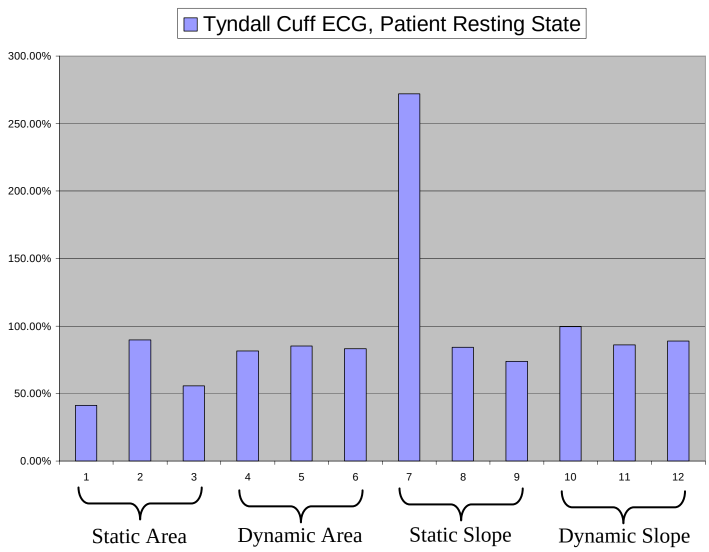


Figure 4.10 (C): An overview of the Area and Slope Techniques Applied to the Tyndall-DMS-Mote ECG Sensor with Power Interference (cf. Table 4.3).

At present the R points in the Tyndall-DMS-Mote ECG reading are the only clearly identifiable regions (cf. figure 4.4 (B),(C)). It does not pick up some of the finer details within a typical ECG signal. The lack of detailed ECG information with the Tyndall-DMS-Mote does not affect the final results, as the current Area and Slope techniques are based on the R-R interval.

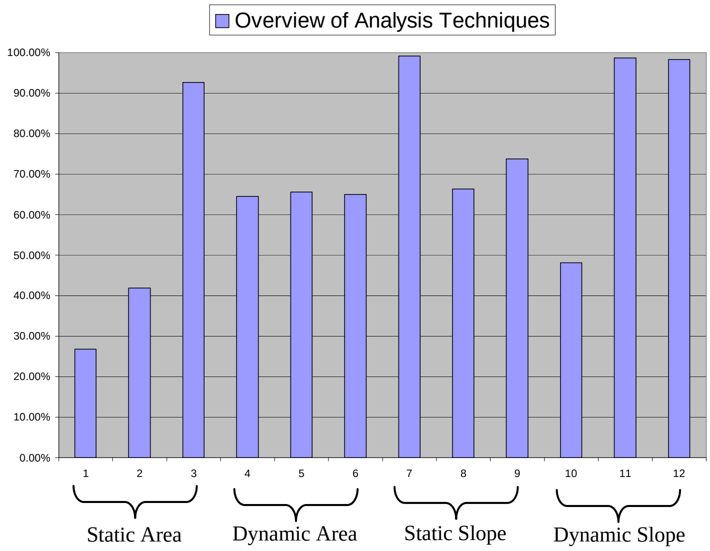


Figure 4.10 (D): An overview of the Average results generated through the application of the Area and Dynamic Techniques. Which were applied to the Power Lab Pulse, Power Lab ECG and Tyndall Sensors (Numbers 1 to 12 represent the analysis technique applied, similar to figure 4.14).

An overview of the Static and Dynamic Area and Slope techniques are presented in figure 4.10 (D). It represents the average of the techniques conducted over the following conditions. Two interference types (power source and physical interaction) and three patient states; 1) Resting, 2) Running and 3) Running to Resting. Finally four Tyndall-DMS-Motes were compared against the Power Lab sensors to identify possible hardware variance which may affect the overall result.

4.4.2 Dynamic Slope Technique under Various Patient Conditions

The manner in which a patient's vital signs are recorded can greatly affect the overall accuracy. An ECG recording is extremely susceptible to external interference. A patient's movement during a recording can interfere with the final sensor reading. The Power Lab and Tyndall-DMS-Mote sensors are designed to record ECG and pulse signals while the patient is in an idle state. Presented in figure 4.11 is an overview of how the three sensors sampled their respective signals. The running signal was recorded immediately after the patient stopped running.

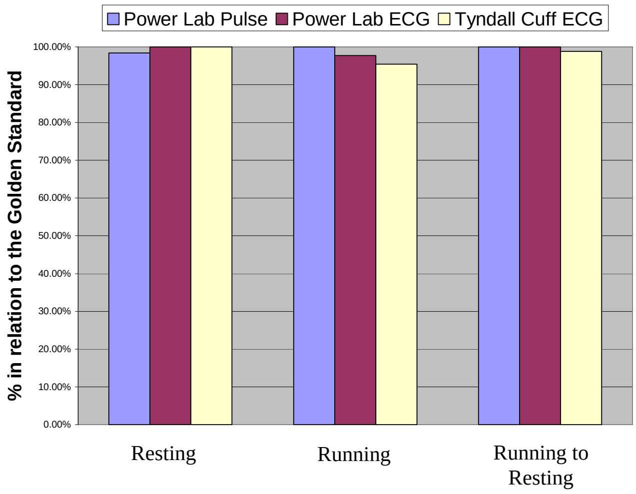


Figure 4.11: Dynamic Slope (average the previous 20 recorded cycles) under three patient states 1) Resting, 2) Running and 3) Running to Resting.

4.4.3 Dynamic Slope Technique and Hardware Variance

An environment or patient sensor is capable of failing and is prone to interference. A single sensor is typically one of thousands mass produced. They are designed to function with built in tolerance levels. Presented in figure 4.12 is a comparison of four Tyndall-DMS-Motes. Cuffs 1 and 4 achieved an accuracy reading of 100% in comparison with the Power Lab Pulse and ECG. Cuffs 2 and 3 achieved an accuracy of 99% and 97% respectively.

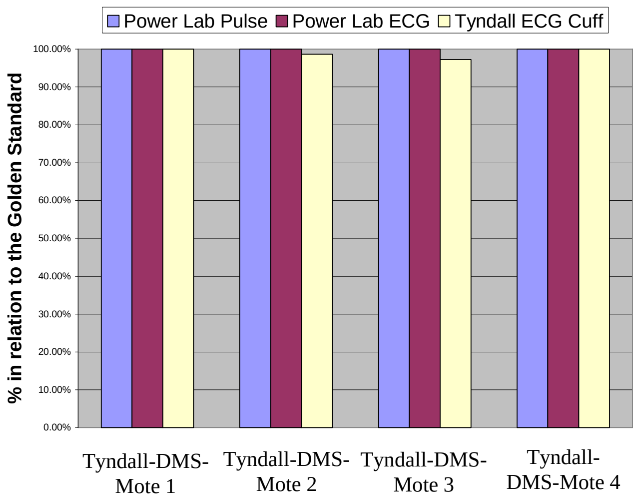


Figure 4.12: A comparison of four Tyndall-DMS-Motes with a Patient State of Resting. A Dynamic Slope (average the previous 20 recorded cycles) was applied.

4.4.4 Transmission of Validated Physiological Signals

It is important to identify as quickly as possible if a sensor is not operating within its tolerance levels. Reacting to a sensor failure in real-time can greatly enhance the overall reliability and dependability. Within a pervasive environment, sensor validation can greatly enhance the quality of data delivered to the end user. Figures 4.13 (A), (B) represent the data delivered to a medical practitioner's PDA. It contains specific patient details with recorded and real time pulse sensor readings. All sensor readings displayed in figure 4.13 (B) represent valid BPM readings. Readings within a DMS tolerance level of 95%. The average accuracy of the two Power Lab sensors and the Tyndall-DMS-Mote ECG is identified as 99.48% for this recorded period of two minutes. Figure 4.13 (C) highlights a scenario where a sensor has not reached the desired level of operation. This is visually represented as a red background to alert the medical practitioner of this fault or error. It is important to note that the two Power Lab sensors reached a level of 100% accuracy. With this in mind the medical practitioner may decide to accept the sensor readings and alert relevant technicians of the Tyndall-DMS-Mote sensor failure. This approach helps to identify faulty sensors within the environment thus improving the overall quality of service and patient care.

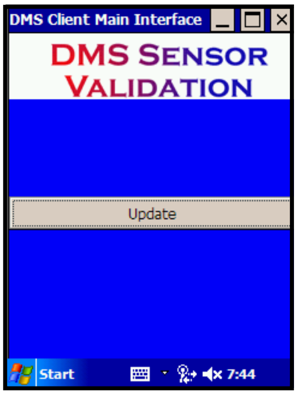


Figure 4.13 (A): Initial DMS-VM Screen.

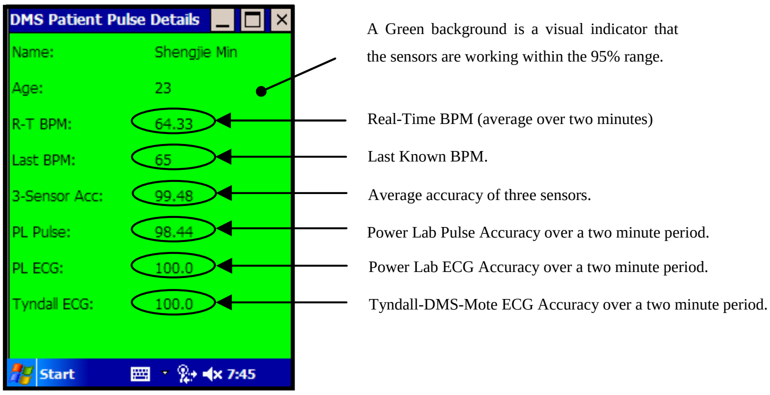


Figure 4.13 (B): Valid Patient Pulse Details with three Pulse reading sources.

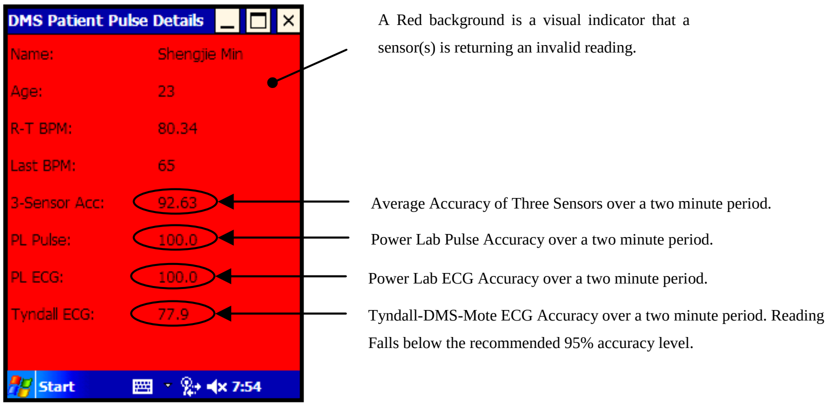


Figure 4.13 (C): Invalid Patient Pulse Details with three Pulse reading sources.

The DMS-VM prototype was tested using the ADInstruments and Tyndall-DMS-Mote biosensors to simultaneously record patient vital signs. These signals are stored and analysed with Area and Slope techniques. Under certain conditions both the Area and Slope techniques have been shown to be effective in identifying the correct BPM number. However the dynamic Slope technique has been demonstrated to be the more versatile under a variety of real world conditions. The Tyndall-DMS-Mote in comparison with ADInstruments medically certified patient sensing devices has been shown to provide the correct patient BPM with a high degree of accuracy. The Tyndall-DMS-Mote allied with the DMS-VM has been demonstrated to maintain this required standard with little to no loss of vital patient datasets.

4.5 Summary

The DMS-VM prototype is designed to validate real-time patient sensor readings. As sensor failures occur within pervasive medical environments, early detection is paramount. A key problem with automated error detection systems are the number of false alarms based on incorrect information. This may result in poor dependability and usefulness. Similar to sensor failure, radio interference/disconnection may generate similar outcomes. The Tyndall-DMS-Mote operating alongside the DMS-VM has been shown to function with a high degree of accuracy, thus demonstrating the potential for remote patient care at the home.

CHAPTER 5

Managing Data Consistency within a Mobile Distributed Environment

5.1 Introduction

Presented in this chapter is the Data Management System-Data Consistency Model (DMS-DCM) component. It is designed to ensure that vital datasets are synchronised between front end devices and backend servers. The DMS-DCM demonstrates that data inconsistency is acceptable if the datasets in question are of little or no importance. The DMS-DCM rules, underline this approach by ensuring that only relevant datasets are captured and synchronised in a timely manner. Therefore high levels of data inconsistency may exist between a number of devices. This helps to save on valuable bandwidth by only transmitting relevant datasets. The results presented demonstrate this fact, as high levels of inconsistency do exist, however medical practitioners only receive the necessary datasets.

5.2 Context Based Data Consistency

Within a pervasive medical environment multiple sources of static and dynamic datasets exist. A high degree of importance is associated with this data, as medical practitioners prescribe relevant patient care based on the information provided. Pervasive environments contain multiple points of access that allow medical practitioners to read and modify patient datasets through PCs, PDAs and other mobile devices. Enabling mobile medical practitioners to modify a patient dataset introduces a new data consistency problem. Presented is the Data Management System-Data Consistency Model (DMS-DCM) [O'Donoghue, 06b]. It is designed to intelligently interact with servers, mobile computing devices and patient sensor nodes within a wireless sensor network (WSN). Effective data consistency is a fundamental requirement within health informatics. It provides the foundation to ensure that medical practitioners receive up-to-date data on time every time. In a distributed dynamic environment multiple views of the same dataset may exist. The DMS-DCM employs a Jade agent platform to ensure that all relevant medical practitioners share a consistent view of patient datasets in real-time.

The Data Management System (DMS) is designed to optimize data management within pervasive medical environments. It is essential within a medical domain that all datasets within the distributed environment (e.g. PCs, PDAs and patient sensors) are up-to-date. Classical data management employ two key operations read and write. In relation to data consistency a write operation can not be executed in isolation. It needs to verify that no other user is interacting with the current dataset and that correct datasets are replicated amongst all users. This is referred to as strong data consistency [Pitoura, 99]. The DMS-DCM applies a similar approach to the view-based consistency of [Huang, 01]. Following this method, data objects of a ‘view’ type are only required to be updated before they are accessed. This ensures that medical practitioners receive the latest datasets upon request. A multicast-based middleware is presented in [Chrysanthis, 03]. In our system Jade agents provide similar data retrieval and dissemination techniques within our pervasive environment. The DMS-DCM is a novel approach in managing data consistency by employing context aware reasoning Jade agents within a data rich pervasive medical environment. The DMS-DCM shares similar qualities to [Cao, 01] where cooperating mobile agents work in unison to ensure data consistency is maintained.

The DMS-DCM architecture is built on two main datasets: core patient data (i.e. patient vital signs, patient history etc.) and context parameters (time, location, profile). By combining these two sets of data in the DMS-DCM model, data consistency techniques may be enhanced. For example consider a patient wearing the Tyndall-DMS-Mote. Pulse rate sensor readings are sampled and transmitted to the DMS-Server. One medical practitioner may update his/her PDA with extra information alongside the current real-time sensor values. Based on the context of the patient, medical staff and the state of the core patient dataset, Jade agents with built-in reasoning may dynamically decide which datasets need to be transmitted to medical devices within the pervasive network. This approach takes advantage of all known real world information in relation to the environment and enhances data management to achieve two key goals: to provide medical staff with relevant data on time and to reduce information overload, thus saving on bandwidth.

The data consistency techniques employed between medical practitioner mobile devices and central medical servers are built using an agent based architectural framework Jade. These techniques enable mobile medical devices to retrieve context relevant information and disseminate real-time datasets to the appropriate medical practitioners and medical servers. The communication links between medical practitioners and the DMS-Server are illustrated in figure 5.1. This diagram highlights a sequence of communication between agents and external patient sensors within a wireless patient sensor network.

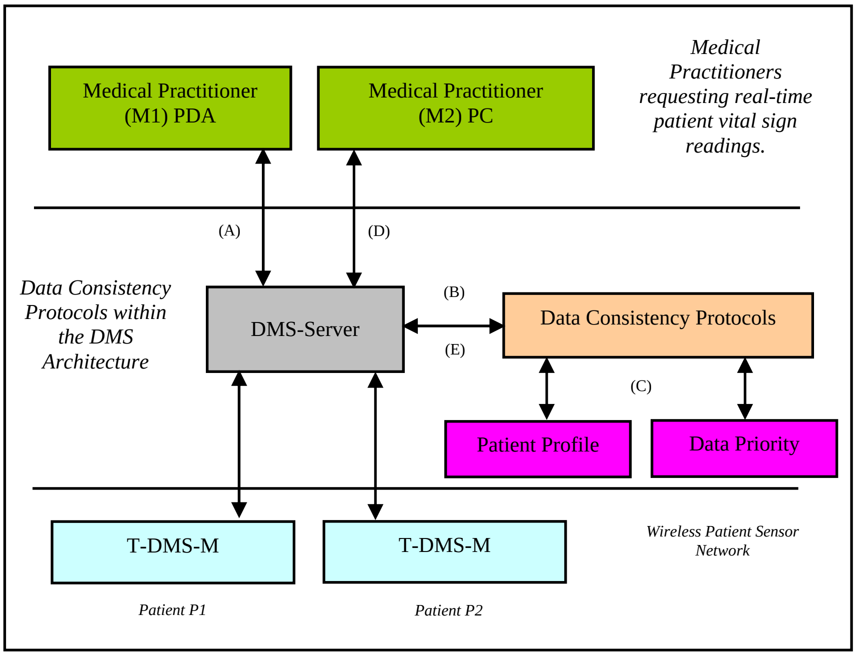


Figure 5.1: DMS-DCM Architecture. (A) A medical practitioner M1 requests patient data from patient P1. (B) Consistency protocols are executed before passing the real-time sensor data onto medical practitioner M1. (C) To save on bandwidth overheads only relevant data is replicated based on the patient profile and sensor data priority. (D) Medical practitioner M2 requests real-time patient data from Patient P1 in parallel to M1. (E) Jade software agents within the DMS-Server replicate all known datasets on M1s PDA and correlate this data with M2 and P1 real-time sensor readings, thus providing a higher QoS. M1 and M2 now share a consistent view of the patient dataset including real-time vital sign readings.

The DMS-Server runs in parallel with DMS-DCM protocols (Jade agents). Here all datasets are prioritised, filtered, and transmitted based on the context of the patient, medical staff and their associated environments. This is achieved by:

- **End User Profile**

A pervasive medical environment may generate large amounts of data. This may result in data overload, poor Quality of Service and potentially, poor patient care.

The introduction of user profiles adds a new dimension to patient data management within a pervasive environment. Presented in [O'Donoghue, 06e] is the DMS-User Profile (DMS-UP). User Profiles are utilised to effectively manage the delivery of relevant information to a practitioner's mobile device, thus improving the QoS. Similar approaches have been applied to overcome network disconnection and/or limited bandwidth [Kambalakatta, 04].

- **Priority**

Data consistency may employ various forms of data replication, where specific or entire datasets are copied to a variety of clients and/or servers. In relation to the DMS-DCM, patient, staff and environment variables are given priorities. This enables the DMS-DCM to update scheduled medical practitioners with relevant real-time information.

- **The Role of Context**

The advantage of integrating a context sensitive data caching technique is outlined in [Mishra, 04]. Here relevant data is cached to neighbouring nodes based on the locality of the mobile user within a wireless network. The DMS-DCM protocols are based on the same philosophy. Merging the profile and priority of patient/staff with the context of the environment (e.g. identity of staff, location and time) decreases the levels of inconsistency as only important datasets synchronised. An outline of some of the key software agents and their interaction between medical staff and specific datasets is presented in figure 5.2. An overview of three of the main software agents is given:

- **Mobile Device Manager Agent (MDMA)**

This is a resident software agent (i.e. JADE) designed to arbitrate between the central DMS-Server and medical practitioner. The agent deals with medical staff requests and incoming server updates. The MDMA interacts with the Mobile Device Data Consistency Agent for pre and post data consistency checks.

- **Mobile Device Data Consistency Agent (MDDCA)**

Incoming and outgoing datasets to and from the mobile device pass through the MDDCA. This agent is responsible for ensuring that not only is the current dataset updated but that all related datasets are resident on the device. It achieves this by interacting with the Central Data Consistency Agent.

5.2.1 Agent Based Data Dissemination

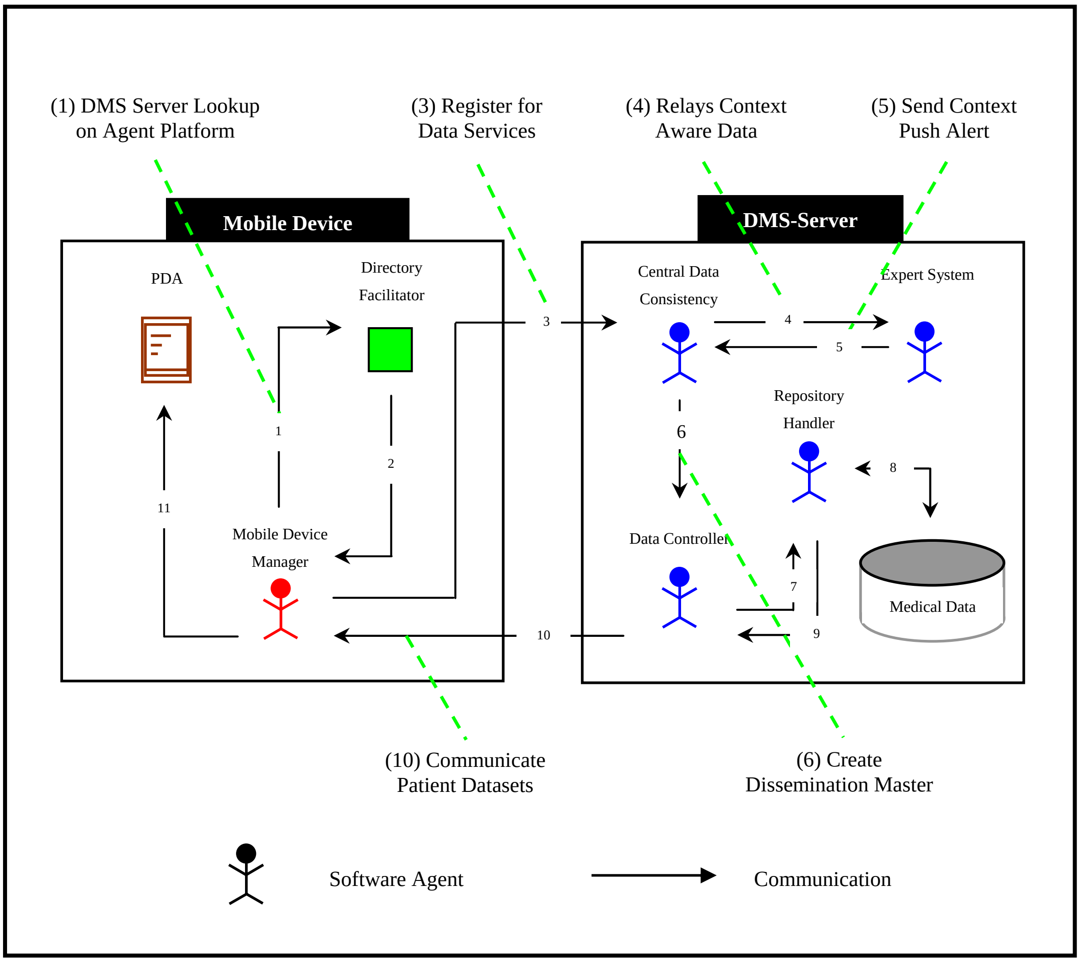


Figure 5.2: The DMS-DCM Agent Based Architecture.

- **Central Data Consistency Agent (CDCA)**

The CDCA manages all real-time sensor data streams and medical practitioner read/write update/requests. CDCA executions are based on the central expert system, which contains a formal set of data consistency rules. All DMS-Server datasets are stored in the Medical Data Store; this includes sensor readings and database records.

5.2.2 Evaluation

The current DMS-DCM prototype has been evaluated for the performance of the agent based data consistency protocols. Experiments were conducted to evaluate the effectiveness of DMS-DCM's broadcasting (server to client) and sampling (based on client requests) capabilities. The primary data source is a patient pulse rate reading over a four hour period. Five data consistency management scenarios are evaluated:

1) Periodic Server to Client Updates. The DMS-Server periodically updates a mobile user (DMS-Client) with patient information datasets.

2) Medical Practitioner's Server Interaction. A mobile medical practitioner requests patient datasets (simulated environment / Poisson distribution).

3) Multiple User Interaction. A single patient may be monitored by more than one medical practitioner. It is important that data concerning that patient is maintained not only on the DMS-Server but also on the relevant medical practitioner's mobile devices.

4) A user profile may contain the risk associated with each patient (i.e. low or high risk patients). This can greatly affect the quantity of datasets which need to be transmitted.

5) Data Priority. The rate at which a patient's sensor reading changes over a period of time can indicate potential patient risks. It may also reduce false alarm generation.

5.2.3 Test Case Environment

All experiments are conducted on patient pulse rate readings in an offline mode. The pulse readings are sampled over a 4 hour period (240 minutes). Sensor readings are stored within the medical database. 240 readings are stored, one reading for each minute.

The definitions of pulse regions in this thesis are generic. For maximum accuracy a set of pulse regions should be designed for each individual patient. This will take into account their medical history, age and level of activity. This will help to decrease false alarm generation and provide the medical practitioners with an accurate patient state of health.

5.2.4 Performance Results

5.2.4.1 Periodic Server to Client Updates

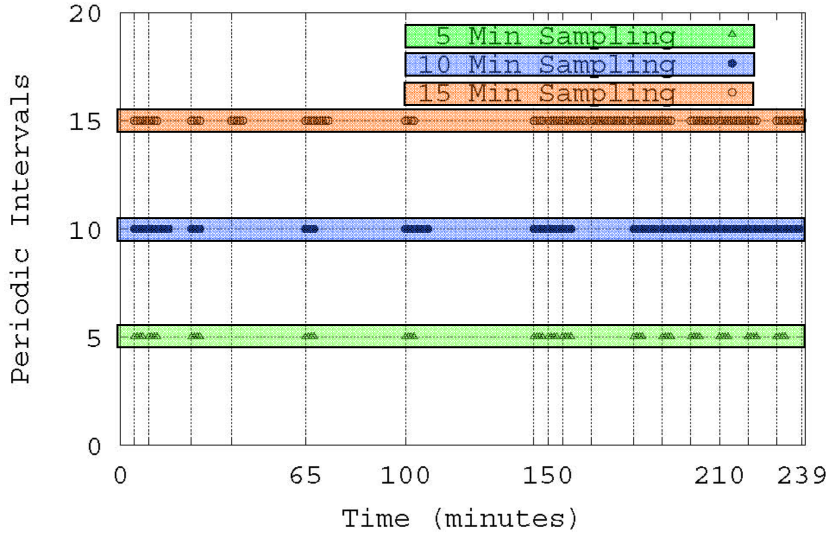


Figure 5.3: Periodic Sampling within a Simulated Environment (Poisson distribution).

The DMS-Server (PC) is configured to periodically transmit a patient's pulse rate every 5, 10 and 15 minutes to a DMS-Client (PDA). The vertical lines (cf. figure 5.3) represent an increase or decrease in the patient's pulse level. The solid horizontal segments during the periodic intervals 5,10 and 15 minutes represent the inconsistency between the DMS-Server and the mobile device.

	Update Interval Period		
	5 Minutes	10 Minutes	15 Minutes
Sensor Activity (Frequency of Change)			
Low (20 Minutes)	10%	35%	40%
Medium (10 Minutes)	20%	70%	64%
High (2 Minutes)	80%	90%	93%
Simulated Environment (Poisson Distribution)	24%	40%	46%

Table 5.1: Percentage of inconsistency during periodic updates.

During a simulation evaluation for 5, 10 and 15 minute periodic updates the mobile device is inconsistent with the DMS-Server for 24%, 40% and 46%

respectively (cf. table 5.1). For a 5 minute periodic update with sensor activities of 20, 10 and 2 minutes the practitioner's mobile device is inconsistent for 10%, 20% and 80% during a 4 hour period. The fundamental issue with periodic updates stems from its lack of context awareness. It does not take into account the meaning of the sensor values and what they represent. If the medical practitioner is to receive the correct data on time every time then a greater understanding as to what each value represents and its context is required.

5.2.4.2 Medical Practitioner Server Interaction

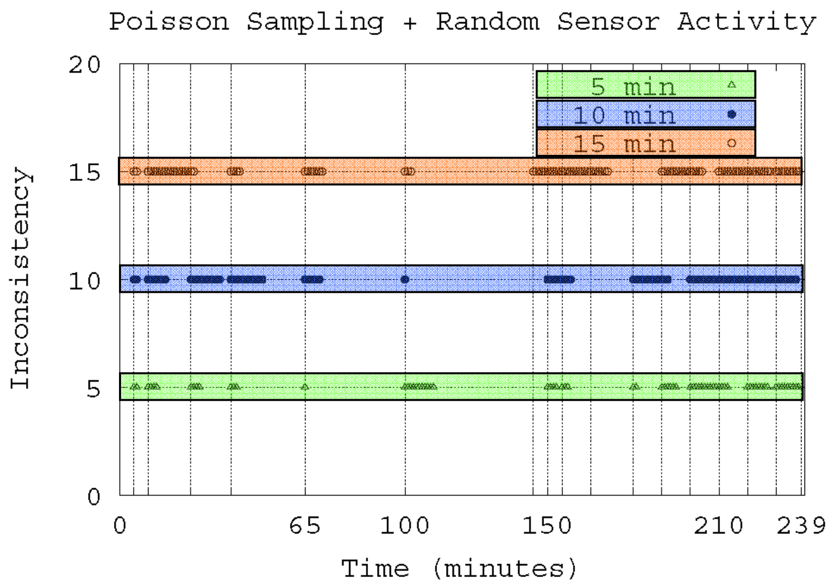


Figure 5.4: Poisson Server Interaction within a Simulated Environment.

A medical practitioner may request patient vital signs randomly throughout the day. Presented in figure 5.4 is the level of inconsistency generated with a Poisson distribution mean of 5, 10 and 15 minutes within a simulated environment (cf. table 5.2).

	Poisson Server Request		
	5 Minutes	10 Minutes	15 Minutes
Sensor Activity (Frequency of Change)			
Low (20 Minutes)	32%	50%	38%
Medium (10 Minutes)	49%	63%	59%
High (2 Minutes)	74%	85%	88%
Simulated Environment (Poisson Request)	30%	41%	43%

Table 5.2: Percentage of inconsistency during Server Interaction.

The percentage of inconsistency of the periodic update and a client request is very high. Both approaches do not evaluate the patient sensor readings against known context elements. If a patient's vital signs were to elevate to a high risk level it is possible that the required data may not reside on the medical practitioner's device. This has the potential to interfere with the level of patient care.

5.2.4.3 Multiple User Interaction

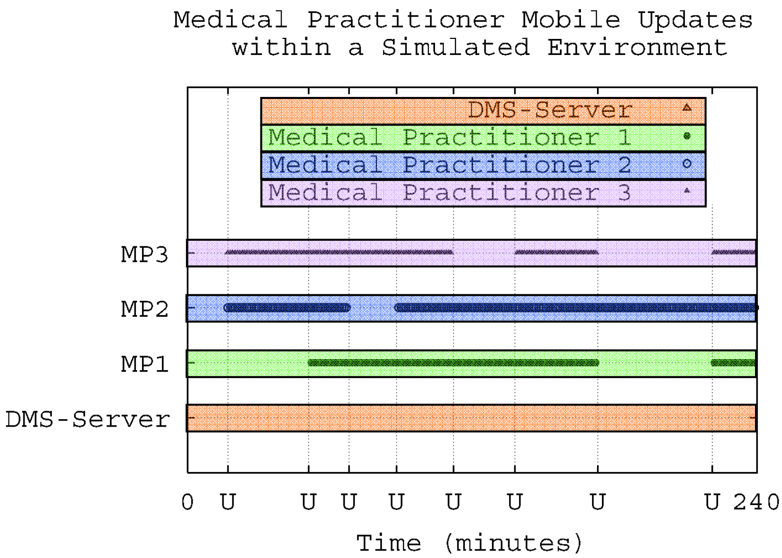


Figure 5.5: Multiple Medical Practitioner Data Updates within a Simulated Environment. U implies a DMS-Server to DMS-Client update. MP1 is medical practitioner 1.

A pervasive medical environment may contain a large number of mobile devices. Each device may hold specific datasets related to a group of patients. To help save on bandwidth and reduce data redundancy only active medical practitioners receive patient vital sign readings when the pulse level reaches an area of risk (cf. figure 5.5, table 5.3). The level of inconsistency is still very high; however the medical practitioners receive all of the patient datasets they require to make a well informed decision. This in turn helps to save on bandwidth usage.

Medical User	Medical Practitioner 1	Medical Practitioner 2	Medical Practitioner 3
Percentage of Inconsistency	58%	85%	61%

Table 5.3: Multi User Interaction.

5.2.4.4 Patient Profile

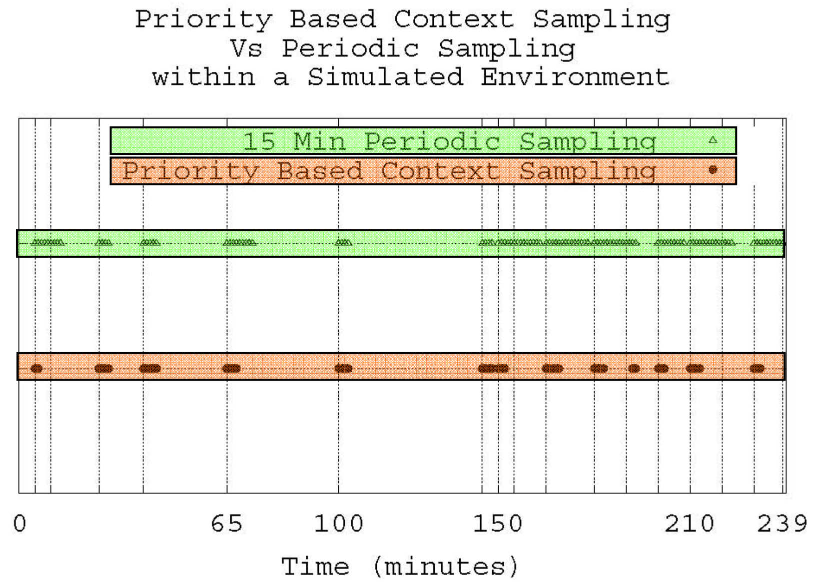


Figure 5.6: Patient Profile Priority Based Update Compared with a Periodic 15 Minute Update. Vertical Lines Represent Sensor Value Change.

The resting pulse rate can differ significantly between an athletic and non-athletic person. A patient's profile containing the type of patient (i.e. athletic or non-athletic) can assist is identifying which datasets need to be transmitted to the medical practitioner for analysis. Presented in figure 5.6 is a comparison between a patient profile priority update and a 15 minute periodic update. With a patient profile priority update more datasets needed to be transmitted. As the patient's pulse rate was below the expected range, the patient priority based approach achieved 25% inconsistency compared with the periodic update of 46%. This approach ensured that the medical practitioner received the necessary data on time to provide better patient care.

The state of a patient allied with the rate of sensor change can significantly alter the required amount of data which needs to be transmitted to a mobile device. Presented in figure 5.7 is a simulated environment with three patient types low, medium and high risk (in relation to their pulse regions (cf. table 5.4)) within a simulated environment.

Age/Risk	Low (BPM)	Medium (BPM)	High (BPM)
0-1	130+-20	100-109 / 151-160	<100 / >160
1-10	100=-30	60-69 / 131-140	<60 / >140
10+	80+-15	60-74 / 96-100	<65 / >100
Athlete	50+-5	40-45 / 55-60	<40 / >60

Table 5.4: Generic Pulse Regions.

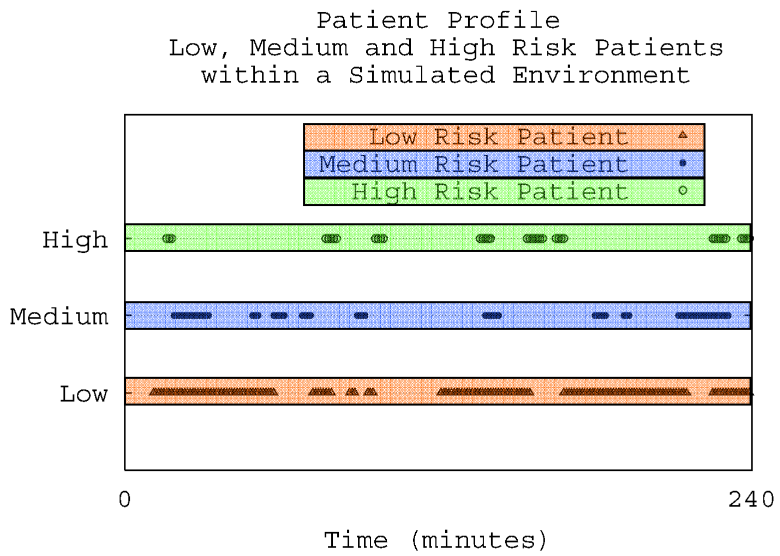


Figure 5.7: Data Transmission with various Patient States based on their level of Risk.

A DMS agent operating within the DMS-Server evaluates the patient's pulse level and compares it against the generic pulse regions. After evaluation it transmits data to relevant medical practitioners based on the following data transmission rules.

Rule	Transmission Time
If pulse reading is within a low risk region.	Once every 60 minutes,
If pulse reading is within a medium risk region.	Once every 10 minutes.
If pulse reading is within a high risk region.	Once every minute.

Table 5.5: Data Transmission Rules.

	Patient Risk Level		
	Low	Medium	High
Percentage of Inconsistency	67%	27%	15%
Number of Client Updates	6	36	114

Table 5.6: Patient Profile with Low, Medium and High Sensor Value Change.

High percentages of inconsistency can be tolerated based on the content of the data. For example in table 5.6 a low risk patient dataset is inconsistent for 67% of the time during a four hour period. The software agent operating within the DMS-Server uses data consistency rules to effectively evaluate the patient state and sensor readings and transmit accordingly. This helps to reduce information overload while, in parallel, frees up resources for higher risk patients whose data needs to be communicated, thus improving the QoS.

5.2.4.5 Data Priority

A patient's pulse level may rise or fall gradually over a period of time. It is therefore necessary to not only read the current value but to view it in the context of previous sensor readings. A new set of data consistency rules are presented (cf. table 5.7) to manage this data consistency requirement.

Pulse Region	Transmission Time
Pulse rate of 70-90 BPM	Once every 20 minutes
Pulse Rate 90-150 BPM	Update 1 every 10 minutes If longer than 10 minutes Update 1 every 5 minutes If longer than 15 minutes Update 1 every 2 minutes If longer than 20 minutes Update 1 every minute
Pulse Rate 150+ BPM	Update Once every Minute

Table 5.7: Gradual Data Consistency Rules.

This new set of data consistency rules were applied within a simulated environment with three patient levels of activity: resting, mildly active and active. The results are presented in figure 5.8 and table 5.8.

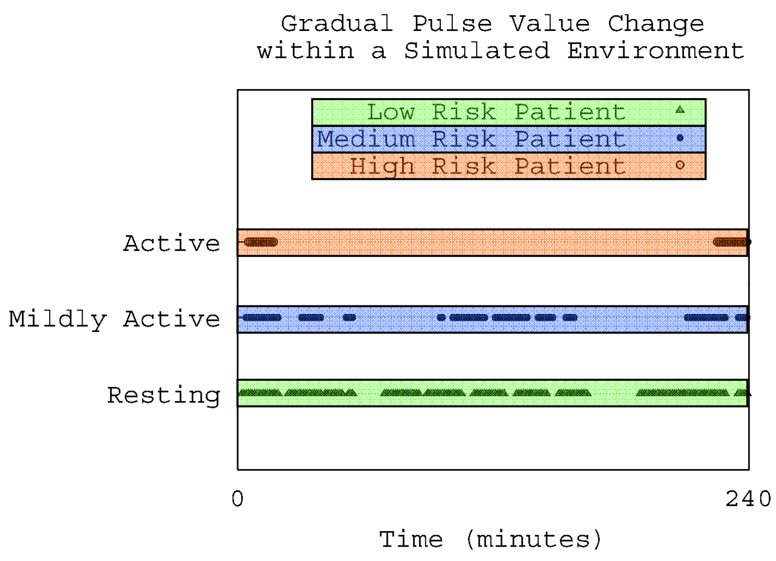


Figure 5.8: Gradual Sensor Reading Change under Various Patient States.

	Patient State		
	Resting	Mildly Active	Fully Active
Percentage of Inconsistency	72%	42%	12%
Number of Client Updates	24	76	208

Table 5.8: Gradual Patient Updates under Various Patient States.

The gradual data consistency rules have prioritised the critical data and transmitted accordingly. The overhead associated with high risk patients can be seen with a total of 208 DMS-Client updates over a four hour period. However this will ensure that the medical practitioners receive real-time datasets at the patient point of care.

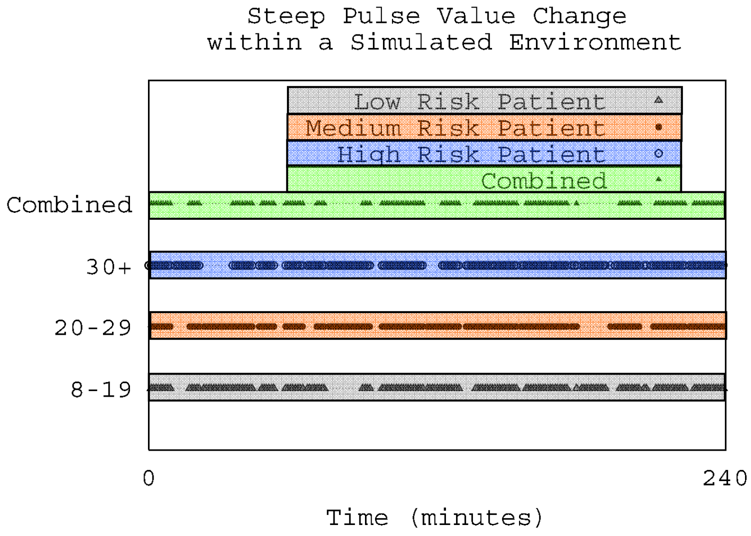


Figure 5.9: Steep Pulse Value Change.

Pulse regions of Interest	8<= X <= 19	20<= X <= 29	X >= 30	Combined
Percentage of Inconsistency	74%	79%	82%	60%
Number of Client Updates	43	35	36	65

Table 5.9: Steep Pulse Value Change. A patient's pulse is monitored over a three minute period. If it increases or decreases by more then or equal to 8 BPM then data is transmitted.

Over a short period of time a patient's pulse rate may increase or decrease dramatically. Such a phenomenon may require immediate attention. Presented in figure 5.9 are the results based on the data consistency rules (cf. table 5.9) applied to a patient's pulse readings over a four hour period. The data consistency rules were applied to a patient in a resting state. These results help to identify peak periods during a patient's daily routine. Through the combination of all three areas of interest a total of 65 updates were passed onto the medical practitioner. This resulted in a 60% inconsistency.

5.3 Conclusion

Presented is the DMS-DCM (Data Management System-Data Consistency Model). It is designed to organise intelligently the large quantities of data within our pervasive medical environments. The DMS-DCM merges context related consistency rules with known context information to deliver relevant real-time information to medical practitioners.

A data source which may reside on a medical practitioner’s mobile device is associated with specific members of the medical staff. This ensures that any context update which occurs within the mobile device is transmitted to key members of staff. The current DMS-DCM prototype demonstrates that issues such as data overload and bandwidth usage may be improved. It also demonstrates that relevant real-time information may be contextually delivered to members of the medical staff, thus improving the QoS.

CHAPTER 6

Knowledge Based Reasoning in the Delivery of Quality Oriented Data

6.1 Introduction

Presented is the knowledge based reasoning component of the DMS architecture. It is implemented using an agent based Data Management System-Tripartite Ontology Medical Reasoning Model (DMS-TOMRM) within a pervasive medical environment. Large quantities of sensory data may be generated within a pervasive medical environment. Using the information gathered, intelligent systems may be developed to assist medical practitioners in real-time. Patient datasets require constant scrutiny and must be analysed in the context of other available information (e.g. patient profile, medical knowledge base). An accurate real-time overview of a patient's state of health is not always possible as communication links may break or servers fail. Therefore it is essential that the information provided on a patient's state of health is taken in the context of available datasets.

Presented is a Data Management System-Tripartite Ontology Medical Reasoning Model (DMS-TOMRM) [Herbert, 06a],[O'Donoghue, 06d]. It is built on three input streams 1) External stimuli (e.g. patient vital signs, patient location), 2) Medical knowledge base (medical database, ontologies) and 3) User profiles (medical history and patient properties). All three sources of information are merged together to provide the medical practitioner with a real-time cardiovascular diagnosis assistant. A key element of the DMS-TOMRM is its ability to cope with physical failures. For example, if the medical knowledge base fails, the DMS-TOMRM may still provide a diagnosis based on the user's profile and current real-time sensor values. This supports the DMS principle of providing a higher quality of service at the patient point of care.

The DMS-TOMRM performs intelligent processing on the available static and dynamic datasets. It correlates this information, highlighting anything of significance (thereby alerting the medical practitioner). The monitoring and handling of data is achieved through an intelligent agent middleware, JadeX [Pokahr, 03]. JadeX provides built-in reasoning and goal oriented facilities which are ideally suited for our context rich medical environments. The relationship and meaning of important data variables within our medical environment needs to be defined. The basis for this careful management of data is the underlying semantic model. A semantic model implies precise definition of the meaning of concepts and terms. Within medical informatics a variety of methods exist which provide precision in biomedical information. These range from medical terminologies, which provide a list of terms with basic hierarchical structures, to very comprehensive representations of medical concepts, encompassing a rich set of relationships between a specific set of concepts. Basic representations may be useful for domains such as information retrieval, while sophisticated representations may be suitable for reasoning. Within a medical environment, different ontologies address specific concerns. HL7 [HL7, 07] is related to medical data exchange terminologies; SNOMED CT [SNOMED CT, 07] is concerned with the precise definition of clinical terms. UMLS [UMLS, 07] addresses the need for “information integration through terminology integration”; it builds on over a hundred source vocabularies and contains over one million concepts. With DMS-TOMRM, the objective is different to that of an ontology model such as UMLS. The goal is to provide a reasoning approach based on all available medical datasets and specifically how we interpret these available datasets in real-time. This leads to a very concise and responsive application-specific ontology, rather than a generic general purpose one.

The computational and communication constraints of an application deployed within a Wireless Patient Sensor Network (WPSN) are directly addressed by the ontology model developed. From this baseline a generic reasoning model may be created to meet application specific requirements. If necessary, detailed ontologies and reasoning models may be integrated into the DMS-TOMRM while maintaining the same application targets. A high-level overview of the current DMS-TOMRM prototype is illustrated in figure 6.1.

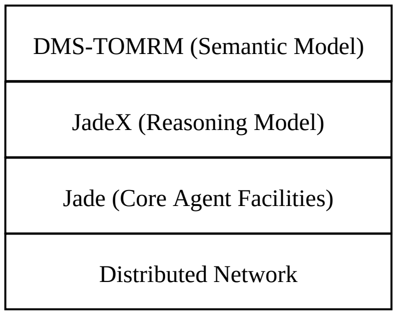


Figure 6.1: A Logical Overview of the DMS-TOMRM Architecture.

The DMS-TOMRM is based on three inputs 1) External stimuli 2) Medical knowledge base 3) User profiles. If one of these input streams should fail, alternative JadeX plans will be activated based on the current context of the patient and their surroundings. The initial DMS version [O'Donoghue, 05] was built on a Jade [Bellifemine, 00] agent middleware, working alongside a Jess expert system and Protégé ontology models. The current DMS version integrates a JadeX adapter. With JadeX a true agent DBI (Desires, Beliefs and Intentions) model is applied. It contains beliefs, goals and plans. Beliefs are all known facts concerning the real-world environment. From this belief base a JadeX agent may activate plans (i.e. intentions) which may contain a set of sub plans (enabling the JadeX agent to react to multiple scenarios.). Finally, the agent desires are specific goals which they aim to complete. When a goal is outlined the JadeX agent will use all known avenues to reach it. The DMS-TOMRM requires a dynamic middleware to cope with intermittent communication links and sporadic end user requests. With JadeX an intelligent agent middleware provides the necessary service (i.e. activate alternative plans/goals on all known contexts).

6.2 Related Work

A number of ambient reasoning models have been developed based on well defined ontologies. How sensors and knowledge bases are integrated plays a pivotal role in their effectiveness [Becker, 04]. The application of ontologies within a distributed environment is presented.

6.2.1 Ontologies

Ontologies within context rich medical environments have been shown to play a key role in providing effective reasoning architectures. In developing a biomedical based ontology model it is important not to view it "as a mere knowledge representation tool" [Ceusters, 03]. It is extremely important that, the true meaning of each data variable is not only understood but is explicit under all related contexts. For example with a partial DMS-TOMRM model a patient's pulse level reading of 120 BPM, may not retain the same meaning if the patient's profile were not available. The term biomedical ontology covers a large range of systems. A number of ontologies have been developed to clearly define specific aspects of a pervasive environment including:

6.2.2 Service Oriented Context Ontologies

As communication links and servers fail, a query injection into a context environment does not always provide the desired result. [Power, 04] describes a context information service which serves ontology-based context queries. It is an important characteristic within a pervasive environment. Well defined relationships between the knowledge base and medical practitioner requirements provide the basis to deliver succinct datasets to the mobile user. Another development within service ontologies is the Web Service Modelling Ontology [WSMO, 07]. Here ontologies are utilised to apply "meaning to all resource descriptions as well as all data interchanged during service usage". [Gu, 04] presents the SOCAM (Service-Oriented Context-Aware Middleware) architecture. It is designed to manipulate and access context aware information. It presents a formal and extensible context model based on the OWL (Web Ontology Language) ontology.

6.2.3 Ontology Based Context Models

A number of general purpose context ontologies have been developed. They outline some of the fundamental aspects required in developing an effective reasoning ontology within our ambient environments. In [Christopoulou, 05] the approach of decoupling application composition from context acquisition and representation is applied. This enables key features of the application to be designed from the top-down, thus providing a pure design approach. The CONON

(CONtext ONtology) is a domain specific ontology built on a hierarchal structure enabling domain specific extensions to be added if required, based on a formal extensible structure [Wang, 04]. As pervasive environments contain multiple heterogeneous devices, development of a common terminology between all participating devices would provide a solid base to develop context reasoning models. In [Alametsa, 04] all devices contain a list of services which it may provide within its ambient environment. A generic ontology for the description of context information was developed. A number of key areas highlighting how ontologies have been applied within our ambient environment were outlined. Development of a complete ontology model which represents every single aspect of our real-world environment is a major challenge. Defining relationships between multiple data sources is extremely complex. Ontologies are most effective when developed for a specific application domain with a well defined collection of datasets and context environment guidelines.

6.2.4 Agent Middlewares and Ontologies

The benefit of deploying intelligent agent middlewares within our medical environments is well documented. They provide sufficient reactive and proactive capabilities to deal with complex, parallel tasks. Scheduling resources for patients within a medical environment is a complex task. Outlined in [Paulussen, 04] is the MedPage architecture which contains Jade agents and JadeX agent adapters. Here agents negotiate with each other based on the current and future states of their respective ambient environments. An Ontology-Driven Software Development in the context of the semantic web is presented in [Knublauch, 04]. Here suggestions are given on how to develop ontology oriented software through Protégé and OWL with an underlining agent platform. Ontologies and intelligent agent middlewares provide the necessary tools in developing effective reasoning domain specific applications. Ontologies may be designed for very specific and complex ambient environments. It is clear that attention to detail, particularly in the area of relationships and context management is paramount in developing a successful ambient application.

6.3 Context Reasoning

Presented is the DMS-TOMRM context reasoning model. It is developed specifically to assist medical practitioners to monitor cardiovascular patients in a non-critical environment. It achieves this through the use of low-disturbance wireless patient sensors (i.e. Tyndall-DMS-Mote). At present, vital signs being monitored include blood pressure, pulse, body temperature and electrocardiogram (ECG). For this application scenario, the ontological model is partitioned to reflect three main sources of information (cf. figure 6.2). This is called the Data Management System-Tripartite Ontology Medical Reasoning Model (DMS-TOMRM). It consists of:

1) Cardiovascular measurement (external stimuli), dealing with the dynamic measurements of the heart and circulatory system as a separate bio-mechanical entity. This model defines how we produce characteristic cardiovascular measurements (for example, pulse rate) from raw sensor datasets.

2) Patient profile record (user profile), providing static information on the individual whose cardiovascular system is being monitored.

3) A medical knowledge base, describing the relevant clinical knowledge and rules in this particular domain. It provides real-time medical based assistance during diagnosis of a cardiovascular patient. It works alongside external stimuli and user profile domains.

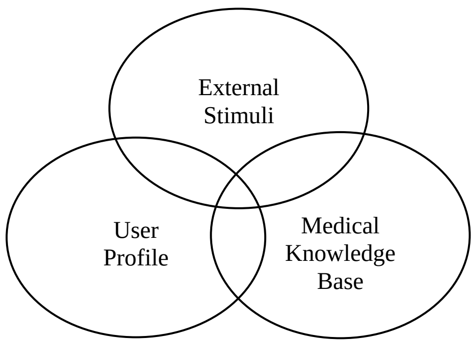


Figure 6.2: The Data Management System-Tripartite Ontology Medical Reasoning Model (DMS-TOMRM).

The overlaps in figure 6.2 of the DMS-TOMRM model indicate that as well as using all three sources of information, we can make use of any two sources to assist medical practitioners. Thus, in the absence of a User Profile, the medical knowledge base could assist the interpretation of the cardiovascular measurements. This may be important in a wireless network environment where not all sources of information may be simultaneously available. An essential task of the DMS is correlating dynamic and static information. The DMS-TOMRM ontology reflects this division of dynamic (external stimuli e.g. sensors) and static (medical knowledge base and patient profile) data. Medical practitioners interact with the DMS-TOMRM through the DMS-Server (cf. figure 6.3). As a medical practitioner examines a patient he/she notes the patient's symptoms on his/her mobile device (e.g. PDA, Tablet PC). This information is then processed by the DMS-TOMRM where it correlates all known information and provides the medical practitioner with a list of known possibilities (diagnosis). This list of possibilities can be rejected or explored further.

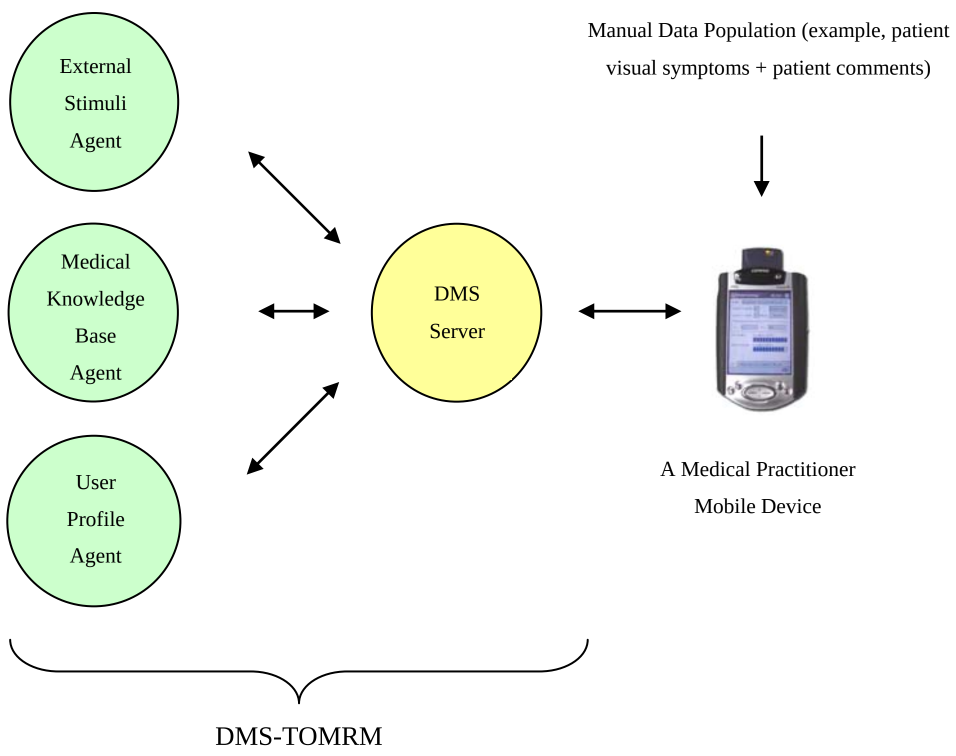


Figure 6.3: Medical Practitioner interacting with the DMS-TOMRM through the DMS-Server.

A brief overview of each of the DMS-TOMRM modules is given:

6.3.1 Medical Knowledge Base

The human pulse rate (in BPM) is taken as an example of a cardiovascular measurement. The pulse rate may be defined as the rhythmical throbbing of arteries produced by the regular contractions of the heart. To fully model a pulse rate and its interaction with other body functions requires a sophisticated and extensive ontology. The current DMS-TOMRM prototype uses BPM readings based on the pulse signals. Apart from defining pulse rate within the medical knowledge base (i.e. specific range, category), patient symptoms and associated causes may be coupled with a specific pulse rate range. This narrows the possible diagnosis list which will be transmitted back to the medical practitioner.

6.3.2 External Stimuli

Patient and ambient sensors return raw data values. In isolation such datasets may assist medical staff in providing a better quality of service. By merging this information with a structured medical knowledge base and user profile, better data analysis techniques may be employed to re-examine the patient's previous state of health. It may also be used to predict future conditions. In relation to pulse rate the Tyndall-DMS-Mote may return BPM values in the range of 0-200 BPM.

6.3.3 User Profile

The medical knowledge base is a collection of facts and relationships concerning specific biological states. This source of information is based on the pure mechanics of the human body (for example, if a patient's pulse rate remains severely elevated over a large period of time, specific organs will begin to fail. This in turn may cause further complications). By merging external sensor values with this medical knowledge base, early detection of such conditions may prevent unnecessary risks. Both the medical knowledge base and external stimuli are valuable sources of information. However they both lack patient specific information including:

1. Patient details, such as age, gender.
2. Current symptoms (e.g. heart palpitations, chest pain).
3. Medical history (operations, family history, allergies).
4. Medication (current and previous medication).
5. Environment context.

This collection of patient specific data is defined as a user profile within the DMS architecture. This valuable set of information greatly enhances the view of a patient's medical condition.

6.4 JadeX and Context Reasoning

To manage our data effectively within a pervasive medical environment a JadeX agent platform is employed. It contains similar qualities to Jade. However JadeX contains a higher abstract level of reasoning. All JadeX agents are given a specific goal which they must reach in order to complete their life cycle. It achieves its goals through a set of plans or through a collection of sub goals. If one of the plans is not feasible (e.g. communication failure), it may activate an alternative plan or a sequence of sub goals. All known facts in relation to the monitored environment are stored within the belief base. A belief base may come in the form of an agent tuple dataset or a local database. JadeX agents react to stimuli in two ways:

1) Internal events may be activated based on a condition trigger within the belief base. Such events activate a deliberation process to see if any other events or plans need to be activated. For example if a patient's pulse level becomes elevated, a JadeX agent will examine all known datasets. If it discovers that the patient is currently running, it may not raise an alarm. If a predefined plan is not currently active then a new plan is instantiated to ensure that the agent's final goal is reached. If the final goal can not be reached then new goals may be adopted.

2) External ACL Messages (Agent Communication Language). Here mobile agents throughout the distributed environment may communicate with each other through ACL messages. Within JadeX all ACL messages must go through a deliberation process. All known goals and plans are identified and activated if in accordance with the belief base. Presented in figure 8.4 is the DMS-TOMRM reasoning model in relation to the JadeX architecture.

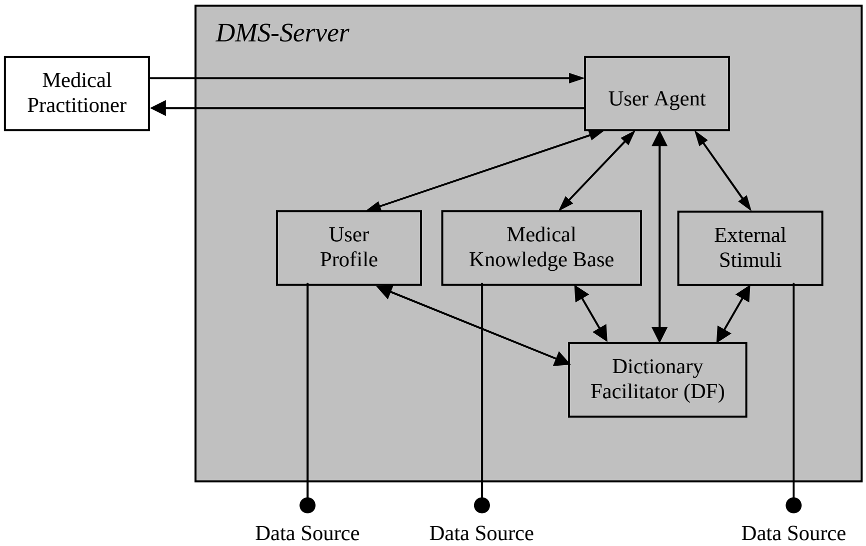


Figure 6.4: An Agent JadeX based DMS-TOMRM Reasoning Model.

Four DMS JadeX agents reside on the DMS-Server, they include:

1) The User Agent manages all incoming and outgoing ACL messages (i.e. requests, informs) from external agents residing on PCs, PDAs. Once initialised the User Agent registers with the Dictionary Facilitator where it may locate other registered agents including User Profile, Medical Knowledge Base and External Stimuli. From here the User Agent is able to access context relevant information. It is responsible for correlating all known data sources (e.g. User Profile, Medical Knowledge Base and External Stimuli) in reaching an overall conclusion regarding the patient's state of health. A new User Agent is created for each mobile user who wishes to interact with the DMS-Server.

2) The User Profile Agent manages all relevant information in relation to a specific patient. All patient data may reside within their mobile device, local database or on the DMS-Server. The security and privacy of this personalised information needs to be examined, but is outside the scope of this thesis.

3) The Medical Knowledge Base (MKB) Agent may come in many forms e.g. local database, ontology, XML document. The MKB stores all known medical facts in a structured manner (Protégé ontology model or Jade Content

Ontology) which enables the MKB agent to filter and locate relevant information in real-time.

4) External Stimuli Agent. A pervasive environment may contain many sensors which transmit vast quantities of information back to a central server. In relation to the DMS-TOMRM model all patient vital sign readings are instantly compared against known facts (i.e. User Profile and/or MKB). This activates a series of sub goals to identify possible irregularities, for example, elevated pulse rate.

6.5 Evaluation

The DMS-TOMRM is evaluated to access how accurately a patient's state may be assessed under various data availability conditions. The effectiveness of the DMS-TOMRM is based on the number of false alarm generations. Three patient types are identified 1) Non-Athletic Adult, 2) Athletic Adult and 3) Child (less then twelve months old). The generic BPM range for each of these three patients in a resting state is presented in table 6.1.

<table>
<tr><th>Patient Type/
State</th><th>Low
Criticality</th><th>Medium
Criticality</th><th>High
Criticality</th></tr>
<tr><td>Non-Athletic</td><td>65-95 BPM</td><td>60-74 BPM ||
96-100 BPM</td><td><65 BPM
>100 BPM</td></tr>
<tr><td>Athletic</td><td>55-65 BPM</td><td>50-54 BPM ||
66-70 BPM</td><td><50 BPM
> 70 BPM</td></tr>
<tr><td>Child</td><td>110-150 BPM</td><td>100-109 BPM ||
151-160 BPM</td><td>< 100 BPM
> 160 BPM</td></tr>
</table>

Table 6.1: Average Pulse Readings for Resting Patients.

6.6 Test Case Environment

Four JadeX agents are developed (cf. figure 6.4). The user profile and external stimuli agents contain their own belief base. The User Agent awaits commands from the user through the JadeX Control Centre (JCC) (cf. figure 6.5). Finally the Medical Knowledge Base agent has access to Jade Content Ontology classes outlining generic pulse regions. After evaluation, the DMS-TOMRM returns a Low, Medium or High patient risk type. These regions are outlined in table 6.2.

Type	Meaning
Low (L)	Expected patient sensor reading
Medium (M)	Slightly outside the expected range
High (H)	Patient needs to be evaluated immediately

Table 6.2: BPM Categories for Patients in a Resting State.

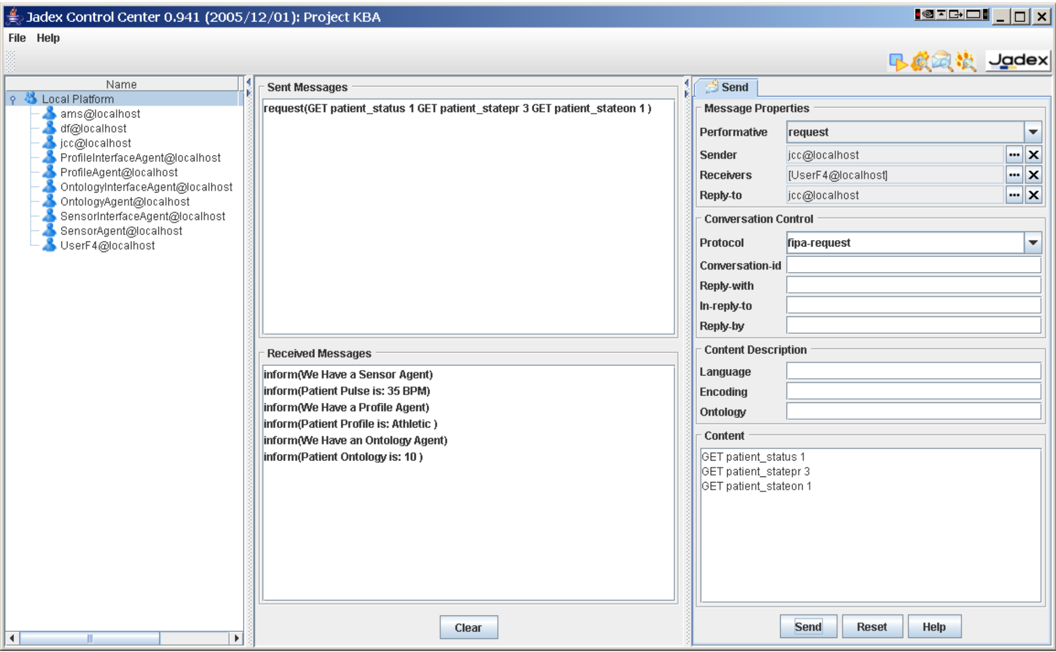


Figure 6.5: JCC interacting with DMS-TOMRM agents.

6.7 Evaluation Results

The DMS-TOMRM is evaluated under two categories partial and complete.

6.7.1 User Profile and Medical Knowledge Base

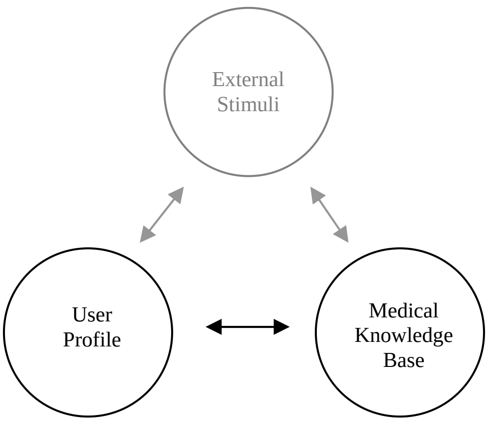


Figure 6.6: Partial DMS-TOMRM, with User Profile and Medical Knowledge Base.

With a partial DMS-TOMRM configuration (cf. figure 6.6) the medical practitioner does not have vital sign sensor readings available. To make a complete assessment of the patient the patient's pulse rate will need to be recorded manually. The medical practitioner will now be able to manually reference this information against the patient's profile and medical knowledge base. This approach can be very time consuming thus reducing the medical practitioner's productivity levels.

6.7.2 User Profile and External Stimuli

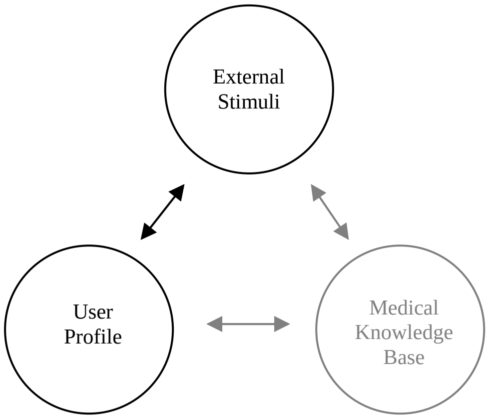


Figure 6.7: Incomplete DMS-TOMRM, with User Profile and External Stimuli.

If access to the medical knowledge base is not available (cf. figure 6.7) the real-time pulse sensor readings may be referenced against the patient profile. This will help to continually monitor the patient's state of health. Also as the patient type is identified the expected pulse range is narrowed. This will help to reduce potential false alarms. In table 6.2, L, M and H regions represent the BPM reading associated with a specific patient state (i.e. Low risk, Medium risk or High risk).

For a partial DMS-TOMRM configuration (cf. table 6.3) one high state was viewed as a medium when it is actually a high (Child). Two high states were viewed as a medium state and one medium state was viewed as a high (Non-Athletic). Finally one false alarm was incorrectly triggered when in fact it was a medium (Athletic). As the medical knowledge base was not available a more precise pulse range was not defined. This resulted in three alarm triggers not activated and two false alarm conditions.

Patient Type/BPM	**35**	**45**	**50**	**56**	**88**	**95**	**125**	**150**	**180**
Child	H	H	H	H	H	**M**	**L**	**M**	H
Non-Athletic	H	H	**M**	**M**	**L**	H	H	H	H
Athletic	H	H	H	**M**	H	H	H	H	H

Table 6.3: The use of a partial DMS-TOMRM model with User Profile and External Stimuli, examines different patient readings (i.e. BPM) under a specific set of scenarios.

6.7.3 External Stimuli and Medical Knowledge Base

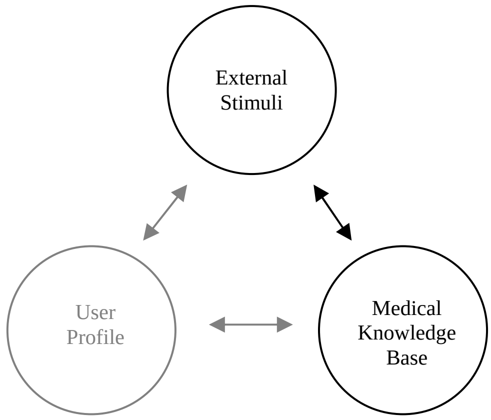


Figure 6.8: Incomplete DMS-TOMRM, with External Stimuli and Medical Knowledge Base.

With a DMS-TOMRM configuration (cf. figure 6.8) the pulse sensor readings are referenced against the medical knowledge base. As the type of patient is unknown to the user agent only the generic pulse ontology and its associated pulse regions is used. In table 6.4 the result for this configuration is presented. Only the non-athletic patient was diagnosed correctly, as it identified two low risk regions. As an athlete resting pulse rate is typically lower then a non-athlete two high risk alarms were raised incorrectly. With a child the resting pulse rate is typically high. Again two high risk alarms were triggered incorrectly. This resulted in the non-triggering of two high risk alarms at the appropriate times.

Type/BPM	**35**	**45**	**50**	**56**	**88**	**95**	**125**	**150**	**180**
Child	H	H	H	H	**L**	**L**	H	H	H
Non-Athletic	H	H	H	H	**L**	**L**	H	H	H
Athletic	H	H	H	H	**L**	**L**	H	H	H

Table 6.4: Partial DMS-TOMRM, External Stimuli and Medical Knowledge Base. Two high risk alarms incorrectly triggered for Child and Athletic. Four high risk alarms were not triggered at the appropriate pulse regions.

6.7.4 Complete DMS-TOMRM Experiment

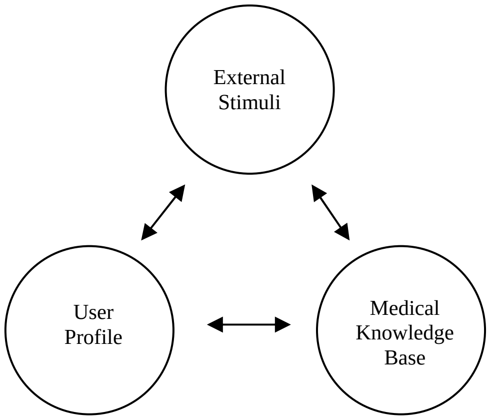


Figure 6.9: Complete DMS-TOMRM, with User Profile, External Stimuli and Medical Knowledge base.

When all information in relation to the patient’s history, type (user profile), medical knowledge base (generic pulse regions) and real-time sensor pulse readings are available a complete picture of the patient’s state of health in relation to their resting pulse rate may be achieved (cf. figure 6.9). In table 6.5 the results for this configuration is presented. All low, medium and high patient states were recognised correctly, resulting in no false alarms. This can help the medical practitioner to build up trust in the system. It also helps to improve the QoS as high risk alarms may be accurately raised when given the correct information.

Type/BPM	**35**	**45**	**50**	**56**	**88**	**95**	**125**	**150**	**180**
Child	H	H	H	H	H	H	**L**	**L**	H
Non-Athletic	H	H	H	H	**L**	**L**	H	H	H
Athletic	H	H	**M**	**L**	H	H	H	H	H

Table 6.5: A Complete DMS-TOMRM Model.

6.8 CONCLUSION

Data management within pervasive medical environment requires an intelligent, sophisticated middleware to combine all known resources in an effective manner. Ontologies may be utilised to define many aspects of our pervasive environment from the very simple definition of medical terms (i.e. pulse range, categories) to the very complex, complete medical reasoning models.

Presented is the Data Management System-Tripartite Ontology Medical Reasoning Model (DMS-TOMRM). This novel approach is designed to assist medical practitioners in providing a higher quality of patient care within a pervasive medical environment. It correlates all known sources of data to assist medical practitioners during diagnosis. A JadeX agent platform is employed to provide the necessary real-time DMS-TOMRM reasoning features.

The DMS-TOMRM component executes a best-effort approach, as a user agent on the DMS-Server will assist medical practitioners during diagnosis with all known data sources. The DMS-TOMRM component gathers and correlates all known information. From here it is able to evaluate the patient's state of health in a coherent data state and provide valuable data to the medical practitioner at the patient point of care. This provides a distinct advantage over current disjointed data gathering approaches, which may be found in the majority of medical systems.

CHAPTER 7

User Context and Data Delivery

7.1 Introduction

Presented is the context based data delivery component of the DMS architecture. It is implemented as an agent based Data Management System-User Context Model (DMS-UCM) within a pervasive medical environment. Software applications operating on a medical practitioner's mobile device should be contextually aware. This can vastly improve the effectiveness of mobile applications and is a step towards realising the vision of a pervasive telemedicine environment. The nature of a medical practitioner emphasises user profile, location, activity and time as key context elements. An intelligent middleware is needed to interpret effectively in real-time and exploit these contextual elements. This architecture is based on an agent based middleware. This framework can proactively communicate patient records to a portable device based upon the active context of its medical practitioner. An expert system is utilised to cross-reference the location and time datasets against a practitioner's work schedule. This dynamic data distribution of medical data enhances the usability of mobile medical devices. The proposed architecture alleviates constraints on memory storage, enhances the use of the handheld device and improves the use of network bandwidth resources. An experimental prototype is presented that demonstrates the usefulness of this approach.

The current DMS-UCM prototype is developed to push context patient datasets onto a medical practitioner's mobile device. The primary context triggers are 1) The medical practitioner's schedule (i.e. which patients are to be examined today), 2) Their real-time location (i.e. ward) and 3) The current time. Presented in figure 7.1 is a segment of a physical ward layout within a typical public hospital. Periodically the medical practitioner's mobile device can transmit its location to the DMS-UCM server. This information is compared with the current time and patient list of the medical practitioner. If a match is found then relevant patient records will be pushed onto the mobile device.

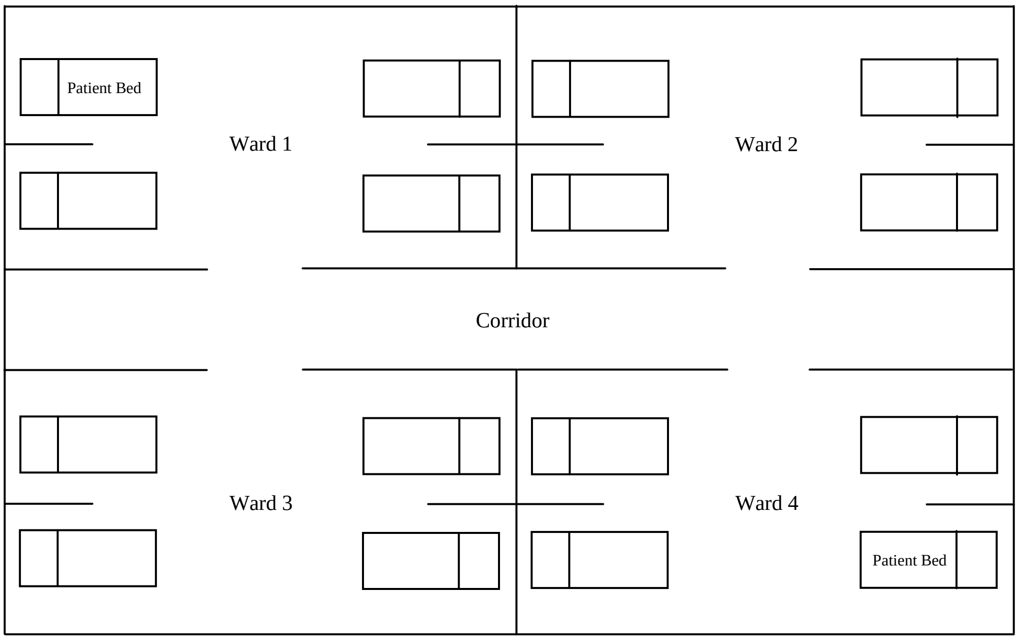


Figure 7.1: Hospital Ward layout.

The storage and visual interface of a portable device affect the usability of a handheld device during analysis of medical records. These limiting factors combined with an intermittent wireless network connection can lead to an unsatisfactory user experience. These issues can be resolved by allowing portable devices to sense and interpret their contextual environment. A context aware mobile medical device can proactively assess its environment's current state. The information gathered from this assessment can be interpreted to determine whether data management operations should be applied to the handheld device (e.g. notify medical practitioner of patient state, transmit relevant data).

This approach anticipates a medical practitioner's specific data requirements. Essentially, relevant patient records are proactively transmitted to a handheld device only when they are required (push based approach). The medical data to be propagated is determined using an informed decision making process that evaluates the contextual environment of the handheld device. This methodology helps alleviate existing problems of information overload and low bandwidth, as only relevant data is transmitted based on the medical practitioner's schedule. The intelligent data management framework enhances the usability and portability of a handheld device.

Additionally, the timely deployment of relevant medical datasets helps to eliminate handheld storage and visual interface constraints. These improvements can lead to increased productivity levels for medical practitioners and help to increase the accuracy of a patient diagnosis. A support infrastructure capable of capturing, communicating and interpreting real time contextual information is necessary for the successful deployment of context aware handheld devices.

7.2 DMS-UCM

The DMS-UCM component proactively communicates patient records to a portable device based upon the active context of a medical practitioner. Agent technology is employed as the enabling middleware within this data management system. The agent based infrastructure facilitating context aware mobile medical devices is shown in figure 7.2. This diagram highlights paths of intercommunication amongst agents as well as dynamic agent creation. Each agent role was determined using an agent oriented analysis and design technique [Wooldridge, 99]. The role of each agent within the UCM framework is outlined as follows:

- Mobile Device Manager

This single instance agent is a permanent resident on the mobile medical device and has responsibility for gathering and maintaining information about the physical device and its owner. The agent operates as the main point of contact between the user and medical applications. The agent registers for a medical record provisioning service. The operation of this service is based upon the contextual environment of the handheld device. The agent is also responsible for informing the provisioning server of any changes in the location of the handheld device.

- Repository Handler

The Repository Handler interfaces with a medical database to obtain patient records.

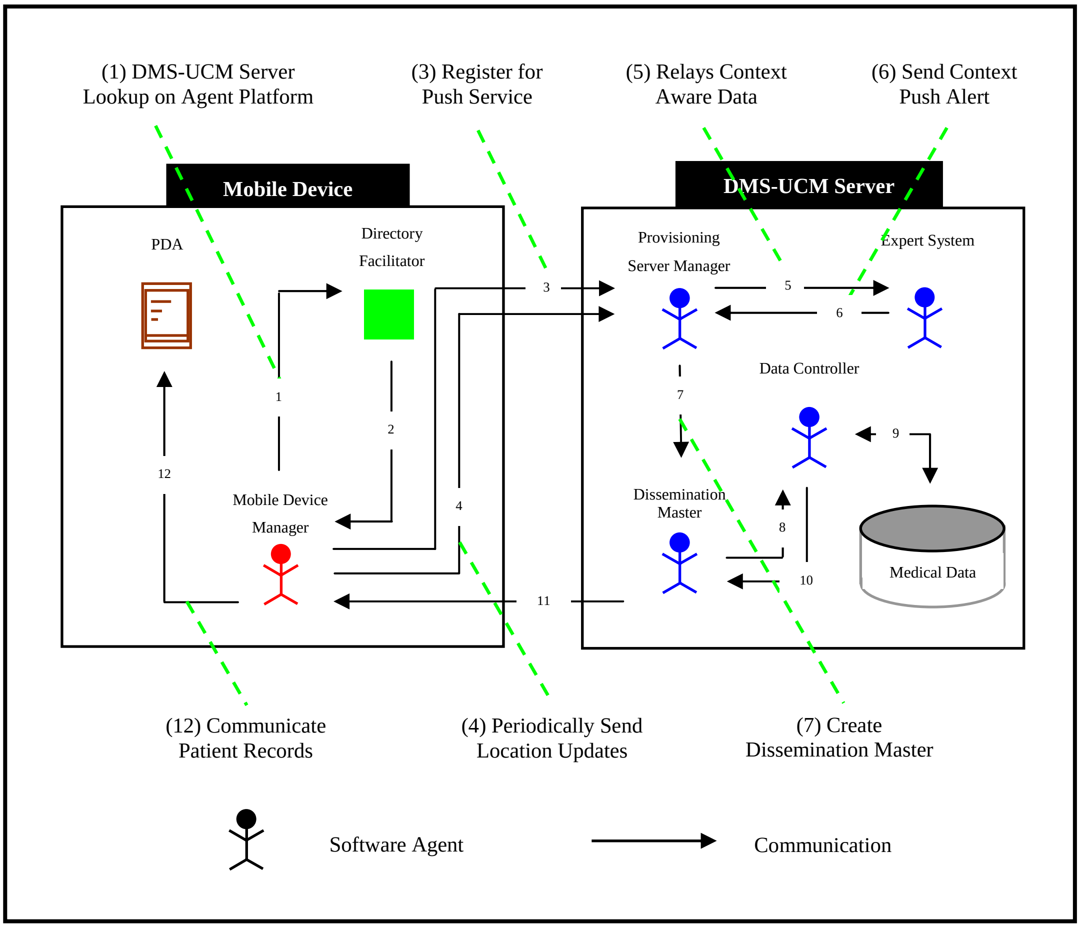


Figure 7.2: DMS-UCM Agent Infrastructure.

- Provisioning Server Manager

This agent is responsible for the provisioning of electronic patient records to handheld medical devices based upon their active context. This agent accepts a request to provide a data management service to a portable device. The Provisioning Server Manager acts upon location updates from medical devices. These location alerts are triggered as the medical practitioner moves within the hospital. This information is communicated to the Expert System Manager to determine whether data configuration is required for the handheld device. A positive response from this agent will result in the creation of a Distribution Master agent to begin propagation of patient records to the mobile medical device.

- Expert System Manager

The Expert System Manager maintains an interface to a rule-based expert system. This agent is responsible for controlling and interacting with the rule engine. This involves gathering the contextual data elements of a handheld device and communicating these values to the expert system. The decision of the rule engine informs the Expert System Manager whether data management operations are required.

- Dissemination Master

This agent is instantiated as needed and is responsible for handling the propagation of patient records to a mobile medical device. This involves efficient inter-communication with the Repository Handler to obtain relevant records from persistent storage. These records are packaged into a medical-based message template and transmitted to the handheld device.

7.3 DMS-UCM System Architecture

The DMS-UCM component facilitates the proactive communication of patient records to a portable device based upon the active context of its medical practitioner. The system architecture of DMS-UCM is shown in figure 7.3.

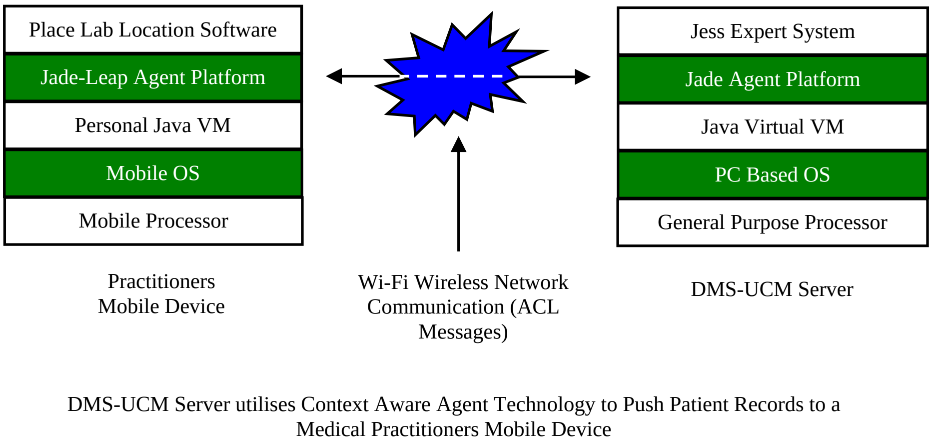


Figure 7.3: DMS-UCM System Architecture.

7.4 Location System

The known real-time location of a mobile device is a required contextual element for the successful deployment of the proposed architectural framework. This is facilitated within the framework through the incorporation of Place Lab technology within each portable device. This is an open source development project that uses a radio beacon based approach for location [LaMarca, 05]. An agent executing on a portable device can use the Place Lab component to estimate its geographic position. This is achieved by listening for unique identifiers (i.e. MAC addresses) associated with each Wi-Fi router. These identifiers are then cross referenced against a cached database of beacon positions to achieve a location estimate. Place Lab was originally designed to estimate location over large geographical areas. Place lab is utilised in this DMS-UCM prototype to identify ward location. This enables the current DMS-UCM framework to associate a Wi-Fi router MAC address with an individual ward. If a patient is moved to another ward then their new ward location will be associated with their patient profile and the medical practitioner's schedule will reflect this change.

7.5 JESS Rule Based Expert System

The deployment strategy to push medical records to a mobile medical device is implemented using a rule based expert system. This informs the decision making process of agents on the DMS-UCM server. Jess is the rule engine and scripting language employed within the framework [Friedman-Hill, 03]. This is a Java based expert system that can interpret and evaluate the contextual elements of a portable device to recommend data management operations. The contextual elements required to enable effective configuration management are the location of the mobile device, the time of day, and the activity of the user. The user activity is derived from a predetermined schedule of practitioner patient appointments. The Jess rules deployed within the current DMS components are trivial in nature that processing time was insignificant and did not merit further investigation as part of this research.

The contextual elements are examined by the expert system through firing a collection of pre defined rules. An example rule which cross references the time aspect of a practitioner's schedule against the current time is shown in figure 7.4.

```
;;Checking For Positive Time Match
(defrule timeChecker1 (ActiveContext (activeStartTime ?activeStartTime))
                      (ActiveContext (activeEndTime ?activeEndTime))
                      (ActiveContext (currentTime ?currentTime))
                      (test (>= ?currentTime ?activeStartTime))
                      (test (<= ?currentTime ?activeEndTime))
                      =>
                      (printout t "TIME_MATCH_FOUND In Rule Base:
                      Current Time is: " ?currentTime"
                      and this is within the appointment
                      start time of: " ?activeStartTime"
                      and the appointment finish time
                      of: " ?activeEndTime "  " crlf)
                      (store TimeOutcome TimeOutcomeMatch))
```

Figure 7.4: A Jess rule for a context based event cross references a practitioner's appointment times with their current time.

7.6 Evaluation

An experimental prototype has been implemented to evaluate the performance of the DMS-UCM component. This prototype facilitates the proactive context based data dissemination of patient records to a portable device, based upon the current context of its medical practitioner. Screenshots of this prototype are shown in figure 7.5 (A), (B) and (C). Figure 7.5 (A) shows the graphical interface displayed to a medical practitioner upon initialisation of the DMS-UCM application. This screen displays the current time and location of the handheld device. The graphical interface is displayed upon receipt of a push of medical records from the provisioning server. This data management operation is triggered by the active context of the medical practitioner.

The propagated data consists of details related to the current practitioner's appointment and any associated patient records as shown in figure 7.5 (B). The graphical interface displays the location and time specific details related to the appointment and a list of associated patient names. The screenshot shown in figure 7.5 (C) is generated upon the selection of a patient name from this list. The graphical interface displays the medical records of the selected patient. It includes general patient information and a list of their health diagnostics. The screen also informs the practitioner of any recent medical scans or sensor readings.

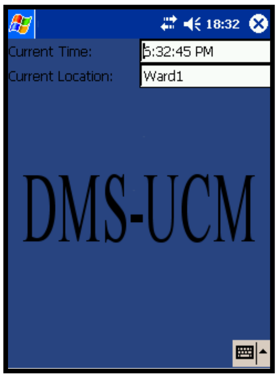


Figure 7.5 (A) DMS-UCM Initial Screen.

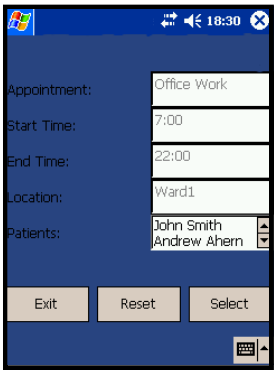


Figure 7.5 (B) Context Aware Patient List.

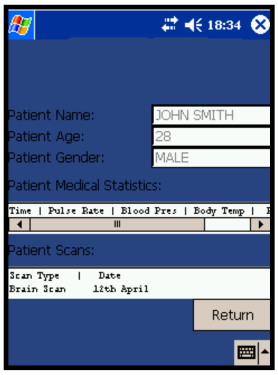


Figure 7.5 (C) Individual Patient Details.

A Place Lab software plug-in resides on each mobile device enabling a location estimate to be communicated to the DMS-UCM server. Agents communicate between the distributed components over a Wi-Fi network. The test case deployment entailed assessing DMS-UCM mobile devices within a laboratory environment. The testing scenario closely emulates the physical ward layout of Cork University Hospital.

Four individual tests were executed to evaluate the performance of DMS-UCM and these are outlined in Table 7.2. Each test was conducted using both the DMS-UCM framework and a Remote Method Invocation (RMI) medical based implementation.

Ward Number	1	2	3	4	5	6	7	8	9	10
Number of Patients	3	2	4	3	2	4	3	3	2	1

Table 7.1: Patient to Ward Distribution.

Type	Test Name	Description
Physical Constraint Test	Handheld Device Storage	DMS-UCM Determine the storage cost on the handheld device resulting from the propagation of patient records. Remote Method Invocation Determine the storage cost on the handheld device resulting from a retrieval of patient records.
	Network Bandwidth Usage	DMS-UCM Determine the network bandwidth consumed by a UCM handheld device. Remote Method Invocation Determine the network bandwidth consumed by the RMI implementation.
Usability and Interaction Test	Data Transmission Time	DMS-UCM Determine the time taken to perform a data management operation. Remote Method Invocation Determine the time required for a retrieval of patient records from a DMS-UCM server.
	User Navigation	DMS-UCM Determine the average user time to navigate to a patient medical record. Remote Method Invocation Determine the average user time to navigate to a patient medical record.

Table 7.2: Overview of DMS-UCM Performance Evaluation Tests.

The tests operated within a simulated environment of ten geographically distributed wards. A timing scenario based upon guidelines for medical practitioner consultations was used in the test [BMA, 04]. This British Medical Association report recommended a minimum of fifteen minutes per patient. The use case scenario randomly distributed twenty seven patients over ten wards to represent the daily workload of a medical practitioner. The patient to ward distribution is shown in Table 7.1. A walkthrough of the wards was conducted by ten individuals to achieve results for each test case. The first test examines the storage required by a DMS-UCM enabled mobile device when applying this use case scenario. Storage costs for the RMI implementation were also obtained. The results of this test case are presented in figure 7.6 (A).

7.6.1 Evaluation Results

The prototype environment consists of a Dell Axim PDA with a Pocket PC 2003 operating system. Each PDA is capable of running the JADE-LEAP agent platform using a Personal Java virtual machine called Jeode. The provisioning server operates on a high-end Pentium PC running the JADE agent platform. The Jess rule based expert system resides on the DMS-UCM server. Patient records propagated to mobile medical devices within the hospital are stored in a SQL Server database.

Data storage on the PDA, using the RMI implementation remains constant due to the retrieval of every patient record for the medical practitioner at each ward. In comparison, the DMS-UCM implementation requires on average 80% less storage by retrieving patient records only associated with the practitioner’s active context (i.e. ward). The second test examines the network bandwidth usage of a DMS-UCM enabled mobile device. Bandwidth usage of an RMI enabled device was also obtained. The results of this test case are shown in figure 7.6 (B). The network usage of the RMI enabled device is again constant and is calculated by determining the cost of invoking a remote retrieval of patient records.

In comparison, the bandwidth usage of a DMS-UCM device fluctuates according to number of patient records transmitted and the frequency of location updates. For example, test results for Ward 1 showed the bandwidth usage within the RMI implementation to be approximately 1100 bytes over 10 wards. The DMS-UCM test results for Ward 1 are based upon a series of location updates (right Y-axis) communicated to the DMS-UCM server and the patient records (left Y-axis) propagated to the mobile device. These combined figures show a bandwidth usage of approximately 250 bytes (approx. 190 bytes – patient records, 60 bytes – location updates) highlighting an improvement of over 75% in relation to the RMI implementation.

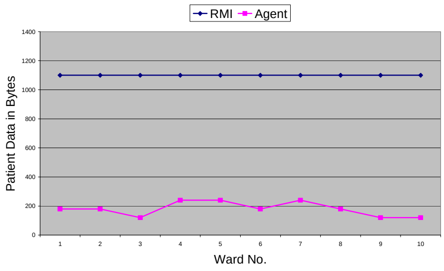


Figure 7.6 (A): Handheld Device Storage.

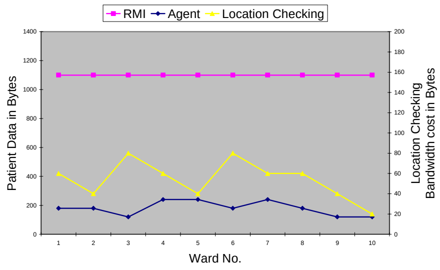
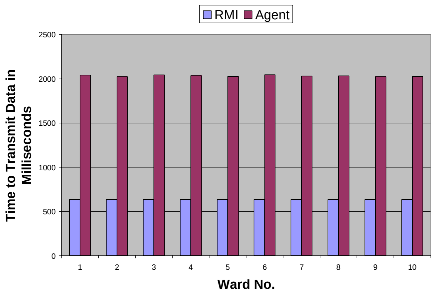


Figure 7.6 (B): Network Bandwidth Usage.

The medical records are currently of a simple textual nature resulting in low memory requirements. Large complex medical records with images of patient scans or numerous patient sensors will show even greater disparity between RMI and DMS-UCM use of network bandwidth usage and mobile device storage requirements. The DMS-UCM component clearly optimises the physical resources of a mobile device.

The third test examines the data transmission time of a DMS-UCM enabled mobile device. Transmission times of the RMI implementation were also obtained. The results of this test case are presented in figure 7.7 (A). The time to communicate patient records within the RMI and DMS-UCM based prototypes is relatively constant. This is mainly due to the stability and availability of the wireless network. The results show the RMI implementation retrieves medical records on average three times faster than the DMS-UCM framework. The primary reason for this disparity is the inherent overhead associated with an agent framework. The fourth test evaluates the average time required by each user to navigate to a specific patient record in each ward. This test case examines the usability of both implementations. The results of this test case are shown in figure 7.7 (B). The concise nature of the patient records returned to a DMS-UCM enabled mobile device showed faster navigation times to individual patient records. The navigation time with the RMI based implementation was on average two seconds slower. The primary cause of this delay is due to the extra time required to locate a specific patient within a larger list. The DMS-UCM implementation clearly improved user interaction by helping to avoid information overload enabling the medical practitioners to source the required information efficiently.



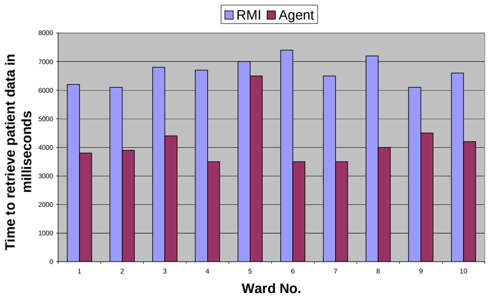


Figure 7.7 (A): Data Transmission Time. **Figure 7.7 (B): User Navigation.**

7.7 Summary

Presented is the DMS-UCM model. A fundamental element of the DMS-UCM is the increased emphasis placed upon the need to interpret intelligently the contextual data elements of a mobile device. The expert system employed within the DMS-UCM framework allows for a comprehensive analysis of these data elements. This enhances decision making ability and enables the framework to deliver more appropriate and proactive support to users of mobile devices. An additional consequence of this sophisticated support infrastructure is its ability to manage acutely the physical device and network resources.

Healthcare organisations are increasing their reliance on mobile links to access patient medical records at the patient point of care. Mobile access to patient records improves the productivity of healthcare professionals and enhances the accuracy of their diagnosis. Mobile analysis of medical records is hindered due to the storage and visual interface constraints of a portable device. These physical constraints affect Data Transmission Time and User Navigation interaction with mobile applications.

The DMS-UCM component is designed to utilise context aware mobile medical devices. The agent based architectural solution proactively communicates patient records to a portable device based upon the active context of a medical practitioner. This distribution of medical data enhances the usability of mobile medical devices as shown in the usability and interaction experiments. The proposed component helps to overcome mobile device and network constraints as demonstrated. The DMS-UCM component is a step towards realising the vision of a pervasive healthcare environment as it provides a meaningful seamless delivery of relevant real-time data to the mobile user.

CHAPTER 8

Distributed Data Management

8.1 Introduction

This chapter deals with the two elements of the Distributed Data Management Component of the DMS architecture. The first element, mobile distributed processing, is outlined in this section. Data communication, within a pervasive environment, must be managed efficiently to save on limited bandwidth resources. For example the Tyndall-DMS-Mote is a wireless sensor device which can monitor patient vital signs non-invasively within and outside a patient's home. A patient's real-time vital sign readings (dynamic data) and archived records (static data) need to be managed, correlated and analysed in a cohesive manner to produce context triggers to alert medical staff. Limited computation is available to clients executing on the sensor node.

The DMS has been developed to process intelligently data generated within the Tyndall-DMS-Mote. Presented is a Mobile-DMS-Client [O'Donoghue, 06c] which executes on a Nokia 9500 Communicator. This client complements the Tyndall-DMS-Mote in its ability to locally process larger amounts of data, thus reducing the need to communicate data to a remote server for computation. When external interaction is required (e.g. to a knowledge base or staff PDA) the DMS can supply information via a context aware agent middleware. Agents effectively encapsulate, extract and interpret real world context aware information ensuring physicians get the "correct" data on time every time. Patient vital sign readings are taken by the Tyndall-DMS-Mote in a less-intrusive non-invasive manner. Details are given on the Mobile-DMS-Client and Tyndall-DMS-Mote prototypes and their ability to communicate and interpret patient blood pressure sensor readings.

Mass produced patient vital sign measuring devices provide a low cost approach for patient monitoring at the home. They are particularly suited in remote areas [WHO, 05] where access to medical assistance is not readily available. They also provide a convenient approach in the self monitoring of non-critical patients at the home. Wireless mobile sensing devices open up new possibilities within the healthcare environment [Winters, 03]. They provide valuable real-time information enabling physicians to monitor and analyse a patient's current and previous state of health. Wireless patient monitoring devices offer an efficient approach in sampling a patient's physiological state. They "provide the opportunity to obtain multiple readings which enable a more accurate estimate of the patient's vital signs" [Pickering, 05]. Through the assistance of telecommunications, patient vital signs may be taken and transmitted over large geographical areas [Zhou, 05],[De Lusignan, 00][Bartan, 06].

To enable remote non-intrusive patient monitoring the Mobile-DMS-Client (i.e. a software agent client executing on a Nokia 9500 Communicator) is introduced. The monitored patient now has the freedom to function as normal within and outside the home, as the Mobile-DMS-Client can transmit over Wi-Fi, Bluetooth (i.e. indoor hotspots) and SMS (Short Message Service) (i.e. wireless outdoor telecommunication services).

Multiple wireless patient monitoring devices exist [Fensli, 05],[Gao, 05] with the capability to communicate wirelessly with mobile devices (e.g. laptops, mobile phones). The Mobile-DMS-Client is built on a Jade-Leap agent middleware. This provides sufficient intelligence to monitor effectively the patient's vital signs without having to interact with the DMS-Server on a continual basis, thus saving on bandwidth. The Mobile-DMS-Client has the added advantage of executing complex tasks locally. This enables computation hungry ECG algorithms to be processed promptly at the patient point of care without having to wait for a response from an external clinical unit. A larger knowledge base (i.e. DMS Ontology, DMS rules) may now reside at the patient point of care as the Mobile-DMS-Client's memory and processing capabilities complement the limited resources of the Tyndall-DMS-Mote.

Non-intrusively Monitoring a Patient's Blood Pressure at the Home

Consider a scenario where an individual, Andrew Smith, suffers from faint dizzy spells and light chest pains. He is admitted to hospital for analysis. It is discovered that Andrew's blood pressure is abnormally high. After a period of medical treatment his blood pressure returns to normal. Andrew's blood pressure needs to be monitored on a daily basis. However as his local doctor is 12 hours away by car, daily checkups are not feasible. To overcome this issue he is given a Tyndall-DMS-Mote and its accompanying Mobile-DMS-Client (i.e. software agent on a Nokia 9500 mobile phone) (cf. figure 8.1). The Mobile-DMS-Client is configured to read Andrew's blood pressure every 30 minutes. Localised blood pressure algorithms can execute to check for potential health warnings. A daily report is generated and transmitted to the DMS-Server enabling physicians to keep a watchful eye during his recovery phase. This requires little interaction from the patient.

Figure 8.1: An Exposed Tyndall-DMS-Mote prototype and Mobile-DMS-Client (i.e. Jade-Leap agent on a Nokia 9500) displaying blood pressure readings and patient details.

8.1.1 Mobile-DMS-Client

Presented in figure 8.2 is the temporal interaction between the Mobile-DMS-Client and the DMS-Server.

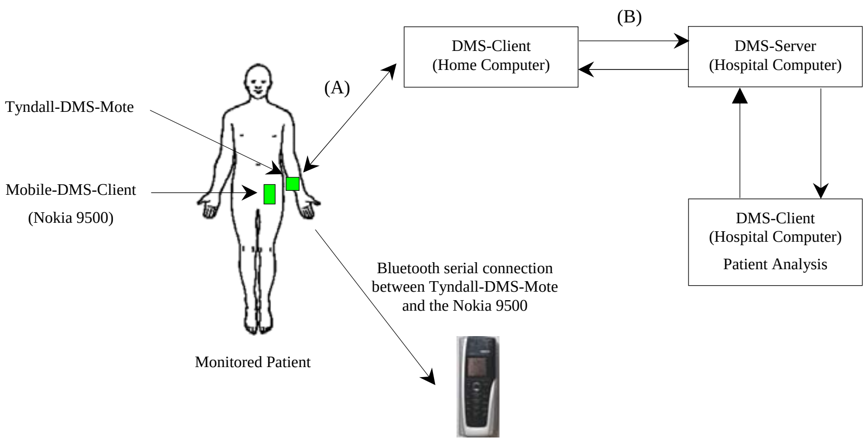


Figure 8.2: Mobile-DMS-Client temporal interaction with the DMS Server. (A) The Mobile-DMS-Client resides on the Nokia 9500; from here it can communicate wirelessly (via Wi-Fi or Bluetooth) with the DMS-Client (Home Computer). Patient vital sign readings taken by the Tyndall-DMS-Mote may be sent directly (via Nordic 2401 radio) to the DMS-Client (Home Computer) or stored on the Mobile-DMS-Client for local analysis. (B) The current DMS prototype communicates via local network.

The Mobile-DMS-Client and the Tyndall-DMS-Mote communicate as follows:

- Tyndall-DMS-Mote (patient sensor) to DMS-Client (Home Computer). Here raw sensor datasets are sent directly (via Nordic 2401 radio) to the DMS-Client for analysis.

- Mobile-DMS-Client (Nokia 9500) to DMS-Client (Home Computer). Data may be sent to the DMS-Client (via Wi-Fi or Bluetooth) for storage, further analysis or as a means to communicate with the DMS-Server. Data may also be processed locally if sufficient medical knowledge resides on the device.

Short term communication failure does not result in data loss. Sensor readings may be stored and processed locally on the Tyndall-DMS-Mote (i.e. 3KB) or the Mobile-DMS-Client (i.e. 2GB). Data compression and filtering techniques may also be applied to save on communication and storage costs. Built-in visual or audio alarms (i.e. warnings) may be activated to inform the patient to seek immediate medical attention.

DMS-Client and DMS-Server

Both the DMS-Client and DMS-Server operate through Jade. It can dynamically manage and organise incoming and outgoing medical data in a context and situation aware manner.

8.1.2 Interacting with the Blood Pressure Sensor

Outlined are three approaches in which the Mobile-DMS-Client may interact with the Tyndall-DMS-Mote in relation to blood pressure analysis:

- **Periodically**

A patient’s profile contains background information on their blood pressure history (i.e. high or low). This dictates the rate sensor readings are taken (e.g. once every 20, 80, 120 minutes). The Tyndall-DMS-Mote may then store these sensor values locally. Patient sensor readings may then be transferred onto the Mobile-DMS-Client on a daily basis, if required.

- **Contextually**

A patient's medical condition (e.g. recovering cardiovascular patient, pregnancy) and current state (active, not active) affect blood pressure regulation. For example during pregnancy a patient's blood pressure may become elevated [Pace, 01] (often referred to as gestational hypertension). If the rule base and ontology systems does not account for this phenomenon, false alarms are sent out to assigned monitors, resulting in a poor quality of service (i.e. incorrect data delivery). Therefore the context of the patient (i.e. current activity) and the situation (e.g. pregnant, thrombosis) play an important role in delivering an efficient service.

- **Custom Run Time Calls**

Custom run-time calls are required for 1) Sporadic adjustment of patient monitoring parameters (e.g. periodic sampling rate) 2) Integration of specialised functions onto the Mobile-DMS-Client through a Jade agent injection and 3) retrieving of patient sensor values.

8.1.3 Evaluation

The current Mobile-DMS-Client prototype has been evaluated for the communication and processing overhead associated with agent based data communication. The primary performance indicator is the time required to process the sensor value. Experiments were conducted to evaluate the effectiveness of Mobile-DMS-Client's processing and broadcasting capabilities. A comparison is made between a single dataset transmitted from the Tyndall-DMS-Mote to the DMS-Client and the Mobile-DMS-Client.

8.1.4 Test Case Environment

One Tyndall-DMS-Mote is utilised to transmit a sensor value. The agent infrastructure is outlined in figure 8.2. The performance indicator time is the average time taken to transmit 20 sensor values in isolation.

8.1.5 Evaluation Results

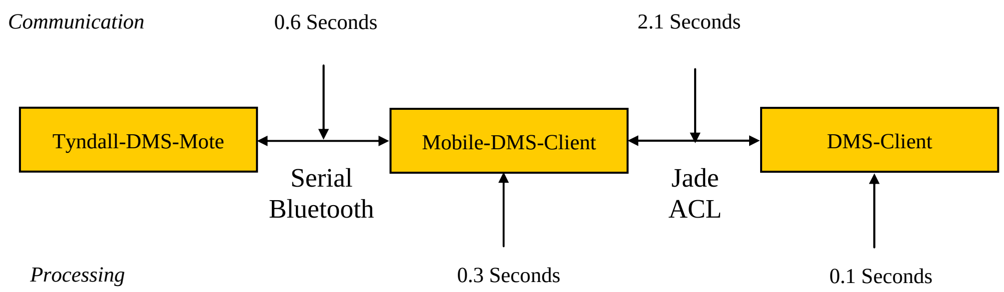


Figure 8.3: Communication and processing overhead associated with the Tyndall-DMS-Mote, Mobile-DMS-Client and the DMS-Client Configuration.

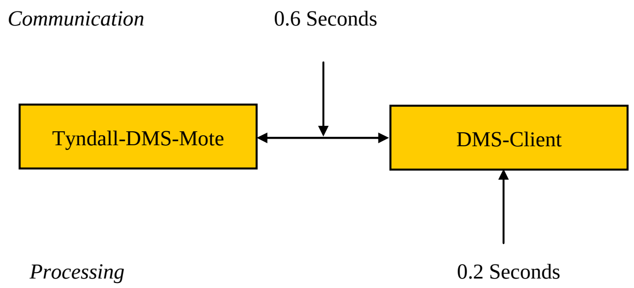


Figure 8.4: Communication and processing overhead associated with the Tyndall-DMS-Mote and the DMS-Client Configuration.

Figures 8.3 and 8.4 illustrate the average time consumed in communicating and processing a single sensor value. The total time from Tyndall-DMS-Mote to DMS-Client is 3.1 seconds. This is substantially higher than the 0.8 (0.6 + 0.2) seconds to transmit and process a signal directly to the DMS-Client via Bluetooth. However this compares favourably with the 0.9 seconds it takes the Mobile-DMS-Client to receive and process a patient vital sign via Bluetooth. For small to medium sized tasks, the Mobile-DMS-Client is comparable in performance with the DMS-Client. Factoring in the advantages of alerting the patient instantly of potential medical risks the Mobile-DMS-Client serves as an effective real-time assistant. For larger tasks, where computation levels are high (i.e. exceed the processing capabilities of the Tyndall-DMS-Mote), it is advantageous to communicate vital sign readings directly to the DMS-Client.

8.1.6 Mobile DMS-Client Conclusion

Presented is the Mobile-DMS-Client and how it monitors an outpatient's blood pressure level in a non-intrusive non-invasive manner. Localised processing and sensing at the patient point of care provides a higher degree of monitoring as it reduces the need to interact with external information servers. The communication and processing overhead, associated with the Jade-Leap agent Mobile-DMS-Client on the Nokia 9500 has been demonstrated under certain conditions to support localised patient care.

8.2 An Agent Middleware Operating with a WSN

The second element of the Distributed Data Management Component of the DMS architecture is presented: WSN agent integration with resource rich backend servers [Herbert, 06b]. Wireless sensor nodes are used to monitor patient vital signs in a medical application. To ensure proper patient care is provided, real-time patient data must be managed correctly in the context of relevant patient information and medical knowledge. Jade runs on resource rich platforms such as servers, PCs. The lightweight Agilla agent platform is designed to run on resource constrained sensor nodes. The Data Management System-Jade Agilla Interface (DMS-JAI) integrates a mobile agent based architecture combining Jade and Agilla. This approach maximises the use of resources contained within the sophisticated agent platform for high-level functionality and the lighter agent middleware for low-level sensor data collection. The resulting system is a unified agent architecture that runs on heterogeneous platforms within a wireless network.

8.2.1 Jade-Agilla Interface

The Agilla middleware executes on TinyOS (an event based operating system designed for wireless sensor networks). Agilla was initially developed for the Mica mote. It has now been successfully ported to the Tyndall-DMS-Mote. Agilla provides the functionality for data processing, communication and supports agent mobility. Agilla provides two types of agent migration: weak migration and strong migration. With weak migration only the code is transferred. For strong migration everything is relocated (heap, program counter, and stack) and the agent resumes execution from its previous node state. An appropriate migration method may be used to respond to dynamic context requirements (patient status, location) thus providing an effective run time solution.

Agent platforms have been applied in a number of medical environments [O'Sullivan, 06] and [Rodriguez, 04]. The integration of software agents is encouraged by the prediction expressed in [Zaslavsky, 04] "that mobile agents capable of discovering, extracting, interpreting and validating context will make significant contribution to increasing efficiency, flexibility and feasibility of pervasive computing systems".

The majority of the components within the DMS are implemented with the more sophisticated Jade agent platform. It is clear that Agilla is appropriate for the important task of data collection from the wireless patient sensor nodes. The basic function of the data collection agent is to retrieve sensor readings from the wireless patient sensor nodes. One may configure the agent to take readings at regular intervals. However during patient monitoring the rate at which these readings are taken may need to be altered. For example, the patient's condition may change or medical staff may require a reading update. The agent paradigm is ideally suited to the reactive/proactive nature of patient monitoring. Therefore extending the agent architecture to the actual patient sensor nodes may yield many benefits.

The concept of integrating a pure agent platform with Agilla was first introduced in [O'Donoghue, 05], with work also reported in [Massaguer, 05]. Implementation of Jade-Agilla integration must fulfil the following minimum requirements: communication via an interface; agent injection from resource rich Jade to the restricted Agilla environment; migration of agents in the sensor network; retrieval of sensor readings by Agilla agents and return of results to the main JADE environment.

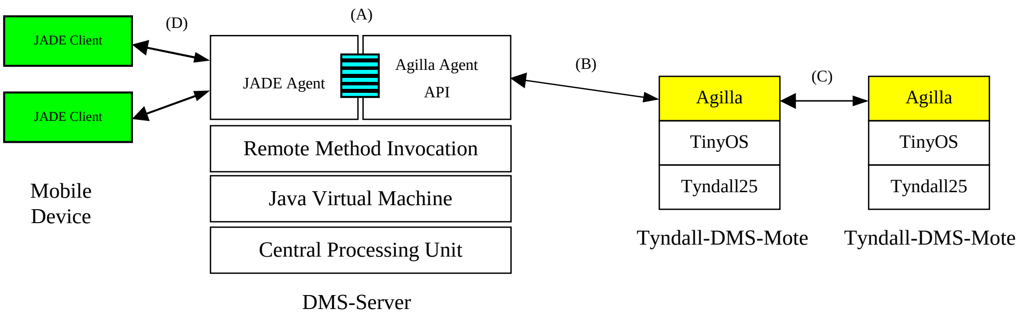


Figure 8.5: Overview of JADE-Agilla Agent Interface. Jade agents operating within the DMS-Server interacting with Agilla agents on a Tyndall-DMS-Mote (A) Jade-Agilla Generic interface, (B) Agent injection. (C) Agent Migration. (D) User update.

Jade to Agilla interaction is achieved via method calls defined in a generic interface (cf. figure 8.5). This provides a straightforward way of injecting Agilla agents. While arbitrary Agilla code can be injected into the wireless sensor network it is useful to have a number of Agilla agents to perform standard tasks such as retrieving sensor readings at regular intervals, or retrieving sensor data in response to an asynchronous request.

Combining agent injection with a library of useful Agilla agents extends the agent architecture to the wireless sensor nodes in a seamless manner. A library of the most common functions has been created. Examples include functions to retrieve patient data periodically and to take an immediate blood pressure reading.

8.2.2 Evaluation

The performance of the DMS-JAI was evaluated based on the response time during a DMS-JAI request. Two scenarios are examined 1) A single request in isolation and heavy network loading and 2) A multi request in isolation and heavy network loading.

8.2.3 Test Case Environment

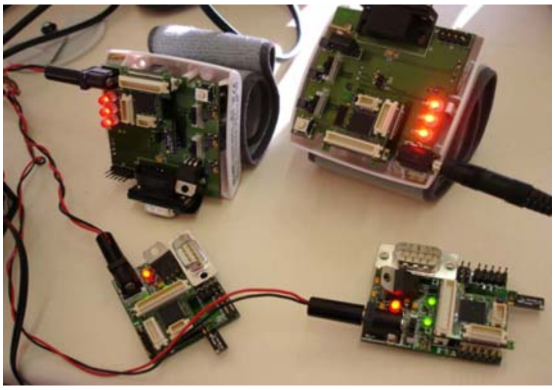


Figure 8.6: Tyndall-DMS-Motes with Agilla Agents communicating wirelessly (Nordic radio).

Four Tyndall-DMS-Motes are utilised. One mote acts as a base station for the WSN. The other three are responsible for gathering sensor values and performing calculations. Three DMS-Clients are created to request sensor values from the Agilla WSN. DMS-JAI GUI and sample Jade code is given in appendix C.

8.2.4 Evaluation Overview

Evaluation of the DMS-JAI is conducted to demonstrate the general performance of the Jade-Agilla integration (cf. figure 8.6). It should be noted that a number of factors affect the performance of a DMS-JAI request. In particular the distance and orientation of the nodes affects their ability to communicate effectively. Agilla agents are executed in order of their queue, First Come First Served (FCFS). Each running Agilla agent is allocated a large time slice to ensure task completion. This can affect the overall time to complete the smallest of tasks. This evaluation of the DMS-JAI is designed to calculate the overhead generated (in time) for Jade agents to communicate and interact with Agilla agents under simple network conditions. Real world factors such as radio interference and sensor failure are not taken into account in these experiments as they extend outside of the scope of this thesis.

8.2.4.1 Simple JAI Scenarios

For a single Jade-Agilla interaction one Jade client (JC1) requests a sensor value to be sampled once every second from Agilla client AC1 (cf. figure 8.7). This experiment was executed in isolation (in that no other Jade or Agilla agents were running in the background). For a multi Jade-Agilla interaction, JC1 requests a sensor value once every second from AC1, JC2 from AC2 once every second, and so on. Results for 20 interactions (JC AC requests) indicate the system performance and the difference between the two experiments (cf. figure 8.8).

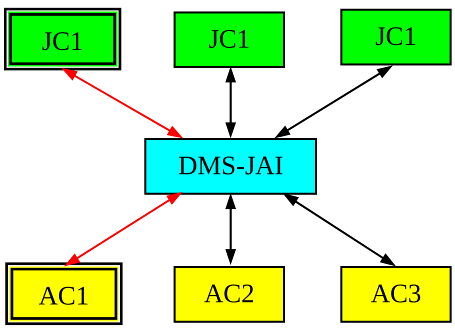


Figure 8.7: Single JADE client (JC1) and single Agilla client (AC1) only. Multiple JADE clients (J1-J3) and multiple Agilla clients (A1-A3) interacting simultaneously.

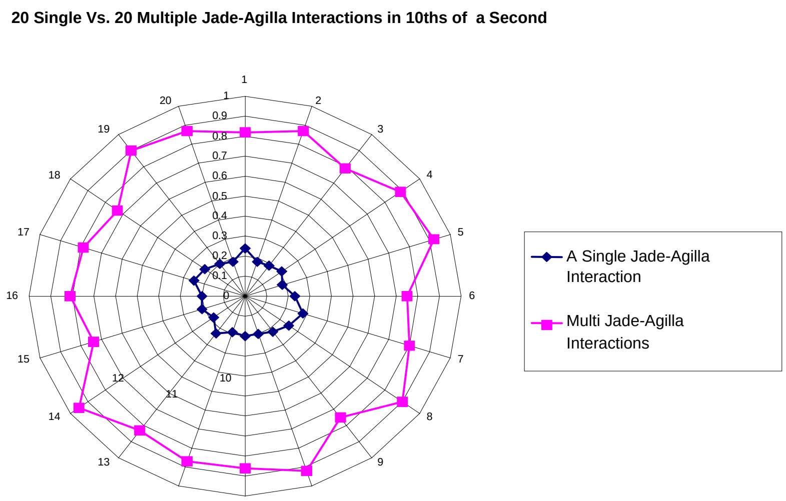


Figure 8.8: Time taken to complete (A) Single JADE-Agilla interactions in isolation (B) 20 Multiple JADE-Agilla interactions in parallel.

During a single Jade-Agilla interaction life cycle 1 sensor reading was returned in 0.2 seconds approximately. In a multi Jade-Agilla interaction this rose to 0.8 seconds. This is attributed to the FCFS task queuing architecture of the Agilla platform and the bottleneck created at the WSN base station.

8.2.4.2 Advanced DMS-JAI Scenarios

Experiment No.	Scenario: Single Sensor Value	Time (Seconds)
1	Single Inject. A Jade agent injects a single Agilla agent into the WSN where it requests a sensor reading from one Tyndall-DMS-Mote.	0.2 seconds
2	Single and Multiple inject in parallel. A Jade agent injects an Agilla agent into one Tyndall-DMS-Mote requesting a single sensor value. In parallel another Agilla agent is injected into the same node requesting a single sensor readings once every 2 seconds	0.6 seconds
	Scenario: Three Sensor Values	
3	Single Inject 2. A Jade agent injects an Agilla agent into one Tyndall-DMS-Mote requesting the three last recorded sensor values.	1.4 seconds
4	Single and Multiple inject in parallel 2. A Jade agent injects an Agilla agent into one Tyndall-DMS-Mote requesting the last three recorded sensor values. In parallel another Agilla Agent is injected into the same node requesting a sensor reading once every 2 seconds.	1.9 seconds

Table 8.1: DMS-JAI Performance Results.

The time to retrieve one data element from a sensor node can vary from 0.2 to 0.6 seconds based on the network conditions outlined in table 6.1. In the request of three sensor values from one node this can fluctuate from 1.4 to 1.9. The major overhead is associated with the communication between agents within the WSN. Bottlenecks are created as Agilla agents need to establish links with neighbouring agents to pass the sensor data back to the base station. At present the identification or location of neighbouring agents is random in nature. Future Agilla versions will employ greater node identification mechanisms to help speed up the data transmissions process.

8.2.5 DMS-JAI Conclusion

Agent based architectures may improve data management within a wireless patient sensor network, and may thus contribute to better patient care. A distributed environment contains both resource rich and resource constrained computing platforms. The lightweight Agilla environment has been ported to the Tyndall-DMS-Mote. It is capable of supporting agents on wireless patient nodes where a more sophisticated but heavyweight agent environment (such as JADE) would not be possible. To maximise the benefits of both the general purpose Jade and specialised Agilla agent platforms, an effective integration was achieved.

The integrated Jade-Agilla environment has been evaluated under a number of scenarios, and is shown to provide a mobile agent infrastructure that extends to the wireless nodes. The integrated architecture supports the agent-based proactive and reactive behaviour demanded by a medical application in a pervasive environment.

8.3 Chapter Summary

In this chapter the Distributed Data Management Component of the DMS architecture was presented. Two elements were examined:

1) Mobile Distributed Processing (Mobile-DMS-Client).
2) WSN Agent Integration with Resource Rich Devices (DMS-JAI).

Within a context rich pervasive environment data needs to be collected, correlated and presented to meet the end user's run time requirements. The utilisation of mobile devices was demonstrated with the Mobile-DMS-Client operating on a Nokia 9500 smart phone (local processing). Vital sensor datasets may be offloaded onto a smart phone for real-time analysis, thus saving on bandwidth and providing the patient with valuable information.

The Agilla platform was ported onto the Tyndall-DMS-Mote. This provided the foundation for the DMS-JAI i.e. a unified agent platform across a range of distributed devices. Jade agents are now able to interact with Agilla based agents within a WSN. The Agilla platform enables agents to be injected into the WSN at run time. This dynamic capability enables context data management facilities to be triggered based on specific environment conditions.

CHAPTER 9

Conclusions

9.1 Introduction

Pervasive environments are complex and dynamic in nature. It is not uncommon for such data rich domains to be made up of disjointed and unorganised datasets. Poor data structures have the potential of degrading the possible benefits of deploying expensive data gathering and mobile devices. If the next generation of pervasive computing environments are to be successful they will require intelligent real-time context aware data management tools. This will help to gather and deliver relevant real-time information to its mobile users.

Presented in this thesis is the Data Management System (DMS) architecture. It is designed to manage the complex datasets generated and stored within a pervasive medical environment. It achieves this through the development of DMS components which collect, correlate and present context related information to the end user in real-time. A *context-sensitive quality data management* architecture can play an important role in correlating the distributed data sources to meet the run time needs of its mobile users.

Sources of data within medical applications include wireless patient sensing devices. A wireless patient sensing device (the Tyndall-DMS-Mote) was developed in collaboration with the Tyndall National Institute and the Department of Medicine (UCC). It contains ECG, pulse, blood pressure (systolic and diastolic) and body temperature sensors. This patient sensing device enabled real world experiments to be conducted. It also helped in the design, development and evaluation of the DMS components.

9.2 Contributions and Results

The primary contribution of this research stems from the development of a Data Management System (DMS) architecture. It is designed to advance the development of *context-sensitive quality data management* for pervasive computing environments. The DMS is built on a number of agent based DMS components which are summarised as follows:

- The validation of real-time patient sensor readings. A direct comparison was made between the Tyndall-DMS-Mote and medically certified patient sensing devices. Presented in chapter 4 is the Data Management System-Validation Model (DMS-VM). It helps to identify and remove erroneous or invalid data elements within a patient vital sign ECG and pulse readings. The DMS-VM demonstrated that mobile lightweight sensing devices in association with relevant data management filtering techniques are capable of achieving comparable results to larger, more expensive standard bedside medical devices. It also demonstrates the potential of monitoring patients at the home in a non-intrusive non-invasive manner.

- An agent based data consistency infrastructure is presented in Chapter 5. The Data Management System-Data Consistency Model (DMS-DCM) utilises all known information (e.g. User profile, medical practitioner's schedule and data priority) to contextually synchronise data residing in multiple sources. A number of data consistency techniques were evaluated. The DMS-DCM is shown to reduce information overload by ensuring that medical practitioners receive contextually relevant information in real-time.

- An intelligent knowledge based reasoning infrastructure, the Data Management System-Tripartite Ontological Medical Reasoning Model (DMS-TOMRM) is presented in Chapter 6. Static and dynamic data sources pertaining to a specific patient may not always be available. Data sources may include user profile (medical history, age etc.), real-time vital sign readings and a medical knowledge base. The DMS-TOMRM is evaluated with respect to possible false alarm conditions. It helps to give value added data as the medical practitioner will be informed as to which data sources were available. This enables a well informed decision to be made and improve the overall quality of patient care.

- An intelligent Data Management System-User Context Model (DMS-UCM) is outlined within Chapter 7. The DMS-UCM provides mobile medical devices with a support infrastructure capable of collecting, communicating and evaluating real-time contextual information. The DMS-UCM component is shown to enhance the usability of mobile medical devices and helps to overcome handheld device and network constraints.

- Local WSN data management capabilities are limited due to their processing and memory limitations. Under ideal circumstances sensory data is transmitted to a central server for processing. However within a home environment such resources are not always available. Presented in Section 8.1 is the Mobile-DMS-Client. It resides on a patient's smart phone. It is capable of collecting and processing sensor readings from the Tyndall-DMS-Mote. The Mobile-DMS-Client demonstrated that mobile devices can play an important role within the next generation of pervasive monitoring environments.

- The WSN Agilla agent platform was successfully ported onto the Tyndall-DMS-Mote. In Section 8.2 the Data Management System-Jade Agilla Interface (DMS-JAI) is presented. It is designed to fully maximise the capabilities of resource rich agent platforms (i.e. Jade) with resource constrained devices (i.e. Tyndall-DMS-Mote) running the Agilla agent platform. The DMS-JAI was evaluated based on its communication and migration capabilities.

- An effective data management infrastructure helps to deliver relevant data to its users in real-time. To achieve this within a complex dynamic environment user profiles help to define data relationships between the user and their environment. The Data Management System-User Profile (DMS-UP) has been utilised in association with the DMS-DCM, Mobile-DMS-Client, DMS-UCM and the DMS-TOMRM. The DMS-UP approach may be utilised to deliver relevant real-time data to the end user on time every time.

Pervasive applications require intelligent context management tools. Software agents have been shown to effectively rationalise context based events within a data rich environment. They are capable of reactive and proactive task execution helping to meet the real-time needs of the mobile users.

9.3 Future Work

The pervasive paradigm is constantly under development. Presented are a number of subject areas which would enhance data management capabilities within a pervasive environment:

- The sensor validation techniques presented in Chapter 4 were evaluated under very strict conditions to reduce potential interference levels. This restricted the monitored patient’s movement. Within a real-world pervasive environment the monitored patient should not be restricted in anyway. To improve on the current sensor validation infrastructure, dynamic filtering techniques need to be developed in association with patient activity sensing devices. Concurrently the Tyndall-DMS-Mote may be redesigned to produce less noisy signals. This may be achieved with the development of the newer Tyndall 10mm module.

- The Agilla agent middleware ported onto the Tyndall-DMS-Mote contains core agent qualities and capabilities. It may be further developed to interact more effectively with its pervasive environment. This may be achieved by increasing its function library to process various environment datasets independently rather than interfacing with backend servers. This will increase its level of autonomy thus becoming an invaluable tool within the pervasive paradigm.

- The DMS-UCM presented in Chapter 7 uses a push-based data management strategy. Relevant patient datasets are sent to the medical practitioner mobile device based on their daily schedule and current location. Emergency situations do arise which may require the immediate attention of specific staff members. Critical patient datasets should be delivered to the relevant medical practitioner in a timely manner. To achieve this, a robust data context infrastructure is required to intelligently pool all information sources. Such an approach may help the medical practitioners provide a higher quality of patient care.

Context-sensitive quality data management architectures can play an important role in utilising the hardware and software resources within a pervasive computing environment. The DMS presented in this thesis highlights the potential of delivering higher levels of data quality to the end user based on their real-world context. Each DMS component helps to eliminate poor data quality factors including faulty sensor readings and data overload. Future pervasive data management architectures with high data quality requirements may employ architectures similar to the DMS in delivering high levels of data excellence.

BIBLIOGRAPHY

[ADInstruments, 07] http://www.adinstruments.com, (2007).

[Alametsa, 04] J. Alametsa, A. Varri, M. Koivuluoma, L. Barna, “The Potential of EMFi Sensors in Heart Activity Monitoring”, 2nd OpenECG Workshop Integration of the ECG into the EHR and Interoperability of ECG Device Systems, Berlin, Germany, (2004).

[Aly, 05] S. G. Aly, “Pervasive Differentiated Services: A QoS Model for Pervasive Systems”, in transactions on engineering, computing and technology, (2005).

[Ancona, 00] M. Ancona, G. Dodero, V. Gianuzzi, F. Minuto and M. Guida, “Mobile computing in a hospital: The Ward-In-Hand Project”, In Proceedings of ACM Symposium Applied Computing, pp. 554-556, (2000).

[Ancona, 01] M. Ancona, et al, “Ward In Hand: Wireless Access to Clinical Records for Mobile Healthcare Professionals”, In Proceedings of the 1st Annual Conference on Mobile and Wireless Healthcare Applications, (2001).

[Anderson, 05] I. Anderson, H. Muller “Towards Qualitative Positioning for Pervasive Environments”, (2005).

[Angove, 07] P. Angove, Brendan O’Flynn, J. Barton, A. Gonzalez, C. O’Mathuna, John O’Donoghue, J. Herbert and S.J. Min, “A Biomonitor Implementation for E-Health", In proceedings of the forth IEEE pHealth conference, (2007).

[Avancha, 02] "Issues in Data Management for Pervasive Environments", In Proceedings of NSF Workshop on Context-Aware Mobile Database Management, (2002).

[Banitsas, 04] K. A. Banitsas, P. Georgiadis, S. Tachakra, D. Cavouras "Using handheld devices for real-time wireless Teleconsultation", (2004).

[Bardram, 07] J. E. Bardram, H. B. Christensen, "Middleware for Pervasive Healthcare", a white paper, (2007).

[Bardram, 04] J. E. Bardram, "Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles", In Proceedings of 2004 ACM Symposium on Applied Computing, pp. 1574–1579, ACM Press, (2004).

[Barkai, 01] D. Barkai, "Technologies for Sharing and Collaborating on the Net", 1st International Conference on Peer-to-Peer Computing (P2P), Linkoping, Sweden, (2001).

[Ballagas, 06] R. Ballagas, J. Borchers, M. Rohs and J. G. Sheridan, "The Smart Phone: A Ubiquitous Input Device" Pervasive Computing, IEEE, Volume 5, Issue 1, pp. 70-77, (2006).

[Bartan, 06] M. C. Bartan, R.W. Jones, "Evaluation of Remote Patient Monitoring in Community Care Homes", Health Computing, (2006).

[Barton, 05] J. Barton, B. O'Flynn, P. Angove, A. Gonzalez, J. O'Donoghue and J. Herbert, "Wireless Sensor Networks and Pervasive Patient Monitoring", Proceedings of Information Technology & Telecommunications Annual Conference (IT&T), (2005).

[Batra, 05] V. S. Batra, N. Batra, “Improving Web Service QoS for Wireless Pervasive Devices”, Proceedings of the IEEE International Conference on Web Services (ICWS), (2005).

[Becker, 04] C. Becker, D. Nicklas, “Where Do Spatial Context-models End and Where Do Ontologies Start? A Proposal of a Combined Approach”, in proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management in conjunction with UbiComp, (2004).

[Bellifemine, 99] F. Bellifemine, A. Poggi, G. Rimassa, “JADE – FIPA Compliant Agent Framework”, In Proceedings of PAAM, pp. 97-108, (1999).

[Bellifemine, 00] F. Bellifemine, A. Poggi and G. Rimassa, “JADE: A FIPA2000 compliant agent development environment”, Proceedings of the fifth International Conference on Autonomous agents, pp. 216-217, (2000).

[Berger, 03] M. Berger, S. Rusitschka, M. Schlichte, D. Toropov, M. Watzke, “Porting Agents to Small Mobile Devices - The Development of the Lightweight Extensible Agent Platform”, In Proceedings of TILAB Journal, (2003).

[Bernstein, 02] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu, “Data Management for Peer-to-Peer Computing: A Vision”, WebDB: Fifth International Workshop on the Web and Databases, (2002).

[Bharatula, 05] N. B. Bharatula, R. Zinniker, G. Troster, “Hybrid Micropower Supply for Wearable-Pervasive Sensor nodes”, ISWC, pp. 196-197, (2005).

[Bisdikian, 02] C. Bisdikian, I. Boamah, P. Castro, A. Misra, J. Rubas, N. Villoutreix, D. Yeh, “Intelligent Pervasive Middleware for Context-Based and Localized Telematics Services”, Proceedings of ACM Mobile Commerce Workshop (MobiCommerce), (2002).

[BMA, 04] “Quality & Outcomes Framework Guidance”, http://www.bma.org.uk/ap.nsf/Content/QualityOutcomes, British Medical Association, (2004).

[Bottazzi, 03] D. Bottazzi, A. Corradi, R. Montanari, “AGAPE: a Location-aware Group Membership Middleware for Pervasive Computing Environments”, In the Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC), (2003).

[Buchholz, 03] T. Buchholz, A. Kupper, and M. Schiffers, “Quality of Context: What it is and why we need it”, In Proceedings of the Workshop of the HP OpenView University Association (HPOVUA), (2003).

[Cannataro, 03] M. Cannataro, D. Talia, “Towards the Next-Generation Grid: A Pervasive Environment for Knowledge-Based Computing”, ITCC pp. 437-441, (2003).

[Cao, 01] J. Cao, A.T.S. Chan and J. Wu, “Achieving replication consistency using cooperating mobile agents” in Proceedings of International Conference on Parallel Processing Workshops (ICPPW), pp. 453-458, (2001).

[Ceusters, 03] W. Ceusters, B. Smith, “Ontology and Medical Terminology: why Descriptions Logics are not enough”, In Proceedings of the conference Towards an Electronic Patient Record (TEPR), (2003).

[Chagoyen, 04] M. Chagoyen and A. Pascual-Montano, “Biological data overload? - Defining today Technologies for Tomorrow”, Proceedings of the ECAI 2004 Workshop on Data Mining in Functional Genomics and Proteomics: Current Trends and Future Directions, pp. 8-11, (2004).

[Choi, 04] N. Choi, S. Venkatesan, R. Prakash, R. Prakash, “A QoS-aware MAC layer protocol forWireless LANs”, Proceedings of International Workshop on Mobile and Wireless Networking (MWN), (2004).

[Christopoulou, 05] E. Christopoulou, C. Goumopoulos, A. Kameas, “An Ontology-based Context Management and Reasoning Process for UbiComp Applications”, Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies, pp. 265-270, (2005).

[Chrysanthis, 03] P. Chrysanthis, K. Pruhs, and V. Liberatore. “Middleware Support for Multicast-based Data Dissemination: A Working Reality”, WORDS, (2003).

[De Lusignan, 00] De Lusignan, A. Althans, S. Wells, P. Johnson, M. Vandenburg, J. Robinson, “A Pilot Study of Radiotelemetry for Continuous Cardiopulmonary Monitoring of Patients at Home.”, Seventh International Conference on Telemedicine and Telecare, (2000).

[Della Mea, 01] V. Della Mea, “Agents Acting and Moving in Healthcare Scenario: A New Paradigm for Telemedical Collaboration”, In Proceedings of IEEE Transactions on Information Technology in Biomedicine, Volume 5, pp. 10-13, (2001).

[Ding, 06] B. Ding, D. Shi, H. Wang, “Towards Pervasive Middleware: A CORBA3-Compliant Infrastructure”, In Proceedings of the 1st International Symposium on Pervasive Computing and Applications, (2006).

[Drugge, 06] M. Drugge, J. Hallberg, P. Parnes, and K. Synnes, “Wearable Systems in Nursing Home Care: Prototyping Experience” Pervasive Computing, IEEE, Volume 5, Issue 1, pp. 86-91, (2006).

[Duran-Limon, 03] H. A. Duran-Limon, G. S. Blair, A. Friday, P. Grace, G. Samartzidis, T. Sivaharan, M. WU, “Context-Aware Middleware for Pervasive and Ad Hoc Environments”, (2003).

[Fensli, 05] R. Fensli, E. Gunnarson, T. Gundersen, “A Wearable ECG-Recording System for Continuous Arrhythmia Monitoring in a Wireless Tele-Home-Care Situation”, Computer-Based Medical Systems, pp. 407-412, (2005).

[FIPA, 04] “Foundation for Intelligent Physical Agents” (FIPA), FIPA Agent Management Specification, available at http://www.fipa.org, (2004).

[Fok, 05] C-L Fok, G-C Roman, C. Lu, "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications" In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS), pp. 653-662, (2005).

[Fontijn, 04] W. Fontijn, P. Boncz, "AmbientDB: P2P Data Management Middleware for Ambient Intelligence", In Proceedings of Perware, (2004).

[Frey, 03] D. Frey, J. Nimis, H. Wörn, and P. Lockemann, "Benchmarking and Robust Multi-agent-based Production Planning and Control", 7th IFAC Workshop on Intelligent Manufacturing Systems, (2003).

[Friedman-Hill, 03] E. Friedman-Hill, "Jess in Action, Rule-Based Systems in Java", Manning Publications, Greenwich, (2003).

[Gao, 05] T. Gao , D. Greenspan, M. Welsh, R.R. Juang and A. Alm, "Vital Signs Monitoring and Patient Tracking Over a Wireless Network" In Proceedings of the 27th Annual International Conference of the IEEE EMBS, Shanghai, (2005).

[Garofalakis, 06] M. Garofalakis, K. P. Brown, M. J. Franklin, J. M. Hellerstein, D. Z. Wang, E. Michelakis, L. Tancau, E. Wu, S. R. Jeffery and R. Aipperspach, "Probabilistic Data Management for Pervasive Computing: The Data Furnace Project". IEEE Data Engineering Bulletin, Volume 29, Issue 1, (Special Issue on Probabilistic Data Management), pp. 57-63, (2006).

[Geer, 06a] D. Geer, “Nanotechnology: The Growing Impact of Shrinking Computers” Pervasive Computing, IEEE, Volume 5, Issue 1, pp. 7-11, (2006).

[Geer, 06b] D. Geer, “Pervasive Medical Devices: Less Invasive, More Productive”, IEEE Pervasive Computing, Volume 5, Issue 2, pp. 85-87, ISSN 1536-1268, (2006).

[Glassey, 03] R. Glassey and R. I. Ferguson, “Modeling Location for Pervasive Environments”, In 1st UK-UbiNet Workshop, (2003).

[Golmie, 05] N. Golmie, D. Cypher, and O. Rebala, "Performance Analysis of Low Rate Wireless Technologies for Medical Applications," Computer Communications, Volume 28, pp. 1266, (2005).

[Gu, 04] T. Gu, X.H. Wang, H.K. Pung, D.Q. Zhang, “An Ontology-based Context Model in Intelligent Environments”. In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, (2004).

[Herbert, 06a] J. Herbert, J. O’Donoghue ”Use of an Ontology for a Wireless Sensor Network Application in a Medical Environment”, Computer Science Department, University College, Cork, February (2006).

[Herbert, 06b] J. Herbert, J. O’Donoghue, G. Ling, K. Fai, C-L. Fok, "Mobile Agent Architecture Integration for a Wireless Sensor Medical Application", In proceedings of First IEEE International Workshop on Intelligent Agents in Wireless Sensor Networks (IA-WSN), Hong Kong, (2006).

[HL7, 07] Reference Information Model http://www.hl7.org/, (2007).

[Huang, 01] Z. Huang, C. Sun, M. Purvis, and S. Cranefield, “View-based Consistency and its Implementation”, In Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the GRID, (2001).

[Huebscher, 04] M. C. Huebscher, J. A. McCann, “Adaptive Middleware for Context Aware Applications in Smart Homes”, 2nd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, Middleware, (2004).

[Indulska, 03] J. Indulska, P. Sutton “Location Management in Pervasive Systems”, In Proceedings of Workshop on Wearable, Invisible, Context-aware, Ambient, Pervasive and Ubiquitous Computing, Conferences in Research and Practice in Information Technology, pp. 143-151, (2003).

[JESS, 07] Jess, the Java Expert System Shell: http://herzberg.ca.sandia.gov/jess/, (2007).

[Kambalakatta, 04] R. Kambalakatta, M. Kumar, S.K. Das, “Profile Based Caching to Enhance Data Availability in Push/Pull Mobile Environments”, In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous), pp. 74-83, (2004).

[Kavimandan, 06] A. Kavimandan, D. Seligmann, R. Klemm, A. John, A. Gokhale, “A Client-Side Architecture for Supporting Pervasive Enterprise Communications”, IEEE Conference on Pervasive Services (ICPS), (2006).

[Kearney, 06] L. Kearney. M. Kidd, M. Miller, J. Sage, M. Khorrami, J. McGee, K. Cassidy and P. Gray, “Utilising Handheld Computers to Monitor and Support Patients Receiving Chemotherapy: Results of a UK-based Feasibility Study”, Supportive Care in Cancer, In Press, (2006).

[Khoukhi, 05] L. Khoukhi, S. Cherkaoui, “A Quality of Service Approach Based on Neural Networks for Mobile Ad hoc Networks”, In Wireless and Optical Communications Networks, 2005, pp. 295-301, (2005).

[Khungar, 04] S. Khungar and J. Riekki, “Context Based Storage: System for Managing Data in Ubiquitous Computing Environment”, Proc. 12th International Conference on Advanced Computing & Communication, (2004).

[Kisazumi, 03] K. Kisazumi, T. Nakanishi, T. Kitasukua, and A. Fukuda, “CAMPUS: A Context-Aware Middleware”, Proceedings of the 6th Joint Workshop on System Development, (2003).

[Knublauch, 04] H. Knublauch. “Ontology-Driven Software Development in the Context of the SemanticWeb: An Example Scenario with Protege/OWL”. International Workshop on the Model-Driven Semantic Web, (2004).

[Krause, 05] M. Krause and I. Hochstatter, “Challenges in Modelling and Using Quality of Context (QoC)”, MATA, pp. 324-333, (2005).

[Kroll, 02] M. Kroll, K. Melzer, H. G. Lipinski, D. H. Grönemeyer, "Accessing DICOM 2D/3DImage and Waveform Data on Mobile Devices", In Proceedings of Second Conference on Mobile Computing in Medicine, Lecture Notes in Informatics, (2002).

[Kumar, 03] A. Kumar, B. Smith, "The Ontology of Blood Pressure: A Case Study in Creating Ontology Partitions in biomedicine", In Press, (2003).

[LaMarca, 05] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild", In Proceedings of Pervasive Conference, (2005).

[Lamberti, 02] F. Lamberti, B. Montrucchio, A. Sanna, C. Zunino, "A Web-based Architecture Enabling Multichannel Telemedicine Applications", In Proceedings of Systemics, Cybernetics and Informatics, (2002).

[Lemlouma, 04] T. Lemlouma and N. Layaïda, "Context-Aware Adaptation for Mobile Devices", International Conference on Mobile Data Management, IEEE MDM Conference, pp. 106-111, (2004).

[Lorincz, 04] K. Lorincz, D. J. Malan, T. R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh and S. Moulton, "Sensor Networks for Emergency Response: Challenges and Opportunities", (2004).

[Lubrin, 05] E. Lubrin, E. Lawrence, K. F. Navarro, "Wireless Remote Healthcare Monitoring with Motes" International Conference on Mobile Business (ICMB), pp. 235-241, (2005).

[Massaguer, 05] D. Massaguer, "Multi Mobile Agent Deployment in Wireless Sensor Networks", Master's thesis, University of California, Irvine, (2005).

[Mazzi, 01] C. Mazzi, P. Ganguly, M. Kidd, "Healthcare Applications Based on Software Agents" Medinfo, pp. 136-140, (2001).

[McCarthy, 86] J. McCarthy, "Notes on formalizing contexts" In Tom Kehler and Stan Rosenschein, editors, In Proceedings of the Fifth National Conference on Artificial Intelligence, pp. 555–560, (1986).

[Meehan, 06] T.P. Meehan, Y. Wang, J.P Tate, M. Curry, A. Elwell, M.K. Petrillo, E.S. Holmboe, "Improving the Quality of Preventive Cardiovascular Care Provided by Primary Care Physicians: Insights from a US Quality Improvement Organization.", International Journal for Quality in Health Care, pp. 1-9, (2006).

[Mihaila, 00] G. Mihaila, L. Raschid, M-E. Vidal, "Using Quality of Data Metadata for Source Selection and Ranking", In Third International Workshop on the Web and Databases, WebDB, (2000).

[Mishra, 04] A. Mishra, M-H. Shin, W.A. Arbaugh, “Context caching using Neighbor Graphs for fast handoffs in a wireless network”, in Proceedings of INFOCOM. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, (2004).

[Mokbel, 04] M. F. Mokbel, X. Xiong, M. A. Hammad, W. G. Aref, “Continuous Query Processing of Spatio-Temporal Data Streams in PLACE”, (2004).

[Nokia, 07] www.nokia.com, (2007).

[Ni Scanaill, 06] C. Ni Scanaill, P. Barralon, G. M. Lyons, N. Noury, “Evaluation of an Accelerometer-Based Mobility Telemonitoring Device in a Smart Home Environment”, In Proceedings of the 4th International Conference on Smart Homes and Health Telematic (ICOST), (2006).

[Norman, 98] D. Norman, “The Invisible Computer”, MIT Press, (1998).

[O’Donoghue, 06a] J. O'Donoghue, J. Herbert and R. Fensli "Sensor Validation within a Pervasive Medical Environment", In Proceedings of IEEE Sensors Conference, (2006).

[O’Donoghue, 06b] J. O'Donoghue, J. Herbert and R. Kennedy "Data Consistency within a Pervasive Medical Environment", In Proceedings of IEEE Sensors Conference, (2006).

[O’Donoghue, 06c] J. O'Donoghue, J. Herbert, P. Stack, "Remote Non-Intrusive Patient Monitoring" in Proceedings of the 4th International Conference on Smart Homes and Health Telematic (ICOST), (2006).

[O'Donoghue, 06d] J. O'Donoghue and J. Herbert "An Intelligent Data Management Reasoning Model within a Wireless Patient Sensor Network", In Proceedings of Artificial Intelligence Techniques for Ambient Intelligence workshop (AITAmI), (2006).

[O'Donoghue, 06e] J. O'Donoghue and J. Herbert "Profile Based Sensor Data Acquisition in a Ubiquitous Medical Environment", In Proceedings of UbiCare 2006 workshop in conjunction with IEEE PerCom, pp. 570-574, (2006).

[O'Donoghue, 06f] J. O'Donoghue and J. Herbert "A QoS Data Management System within a Pervasive Medical Environment", In Proceedings of the Doctor Colloquium in conjunction with Pervasive Conference, (2006).

[O'Donoghue, 05] J. O'Donoghue, J. Herbert, "Data Management System: A Context Aware Architecture For Pervasive Patient Monitoring" in Proceedings of the 3rd International Conference on Smart Homes and Health Telematic (ICOST), pp. 159-166, (2005).

[O'Flynn, 05] B. O' Flynn, A. Barroso, S. Bellis, J. Benson, U. Roedig, K. Delaney, J. Barton, C. Sreenan and C. O'Mathuna, "The Development of a Novel Miniaturized Modular Platform for Wireless Sensor Networks", Proceedings of the IPSN Track on Sensor Platform, Tools and Design Methods for Networked Embedded Systems (IPSN/SPOTS), (2005).

[O’Flynn, 06] B. O'Flynn, P. Angove, J. Barton, A. Gonzalez, J. O'Donoghue and J. Herbert “Wireless Biomonitor for Ambient Assisted Living”, In Proceedings of IEEE ICSES (International Conference on Signals and Electronic Systems), (2006).

[O’Flynn, 07] B. O'Flynn, P. Angove, J. Barton, A. Gonzalez, Cian O’Mathuna, J. O'Donoghue, S.J. Min and J. Herbert “Wireless ECG - Biomonitor for Implementation for E-Health”, In Proceedings of the 4th pHealth workshop, (2007).

[O’Sullivan, 06] T. O’ Sullivan, J. O’Donoghue, J. Herbert, and R. Studdert, “CAMMD: Context Aware Mobile Medical Devices”, International Journal of Universal Computer Science, Special Issue on Pervasive Health Management: New Challenges for Health Informatics, (2006).

[Pace, 01] B. Pace, “High Blood Pressure During Pregnancy”, Journal of the American Medical Association, pp. 1664, (2001).

[Paulussen, 04] T. O. Paulussen, A. Zoller, A. Heinzl, A. Pokahr, L. Braubach, W. Lamersdorf. “Dynamic Patient Scheduling in Hospitals”, Coordination and Agent Technology in Value Networks, publisher GITO Berlin, (2004).

[Paradiso, 05] J. A. Paradiso, T. Starner “Energy Scavenging for Mobile and Wireless Electronics”, Pervasive Computing, IEEE, Volume 4, Issue 1, pp. 18-27, (2005).

[Pawar, 07] P. Pawar, A. van Halteren, K. Sheikh, “Enabling Context-Aware Computing for the Nomadic Mobile User: A Service Oriented and Quality Driven Approach”, In Proceedings of the IEEE Wireless Communication and Networks Conference, (2007).

[Pedersen, 04] K.V. Pedersen, “Context Based Support for Clinical Reasoning”, In Proceedings of the 4th International Conference of Knowledge Management (I-Know), pp. 397–404, (2004).

[Perich, 04] F. Perich, A. Joshi, T. Finin, and Y. Yesha, “On Data Management in Pervasive Computing Environments”, Knowledge and Data Engineering, Volume 16, Issue 5, pp. 621-634, (2004).

[Pickering, 05] T. Pickering, “The Measurement of Blood Pressure in Developing Countries”, Blood Pressure Monitoring, (Editorial), (2005).

[Pitoura, 99] E. Pitoura, B. K. Bhargava, “Data Consistency in Intermittently Connected Distributed Systems” In Proceedings of IEEE Transactions on Knowledge and Data Engineering, pp. 896-915, (1999).

[Pokahr, 03] A. Pokahr, L. Braubach, W. Lamersdorf, “Jadex: Implementing a BDI Infrastructure for JADE Agents”, in EXP In Search of Innovation (Special Issue on JADE), (2003).

[Power, 04] R. Power, D. Lewis, D. O'Sullivan, O. Conlan, V. Wade, "A Context Information Service Using Ontology-based Queries", In Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp), (2004).

[Rao, 91] A. S. Rao and M. P. Georgeff, "Modeling Rational Agents within a BDI-Architecture" In Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, (1991).

[Rodriguez, 04] M. Rodriguez and A. Preciado, "An Agent Based System for the Contextual Retrieval of Medical Information", AWIC Conference, pp 64-73, (2004).

[Rodriguez, 03] M. Rodriguez, J. Favela, V. M. Gonzalez and M. A. Munoz, "Agent Based Mobile Collaboration and Information Access in a Healthcare Environment" in Proceedings of Workshop of E-Health: Applications of Computing Science in Medicine and Health Care, (2003).

[Roussos, 05] G. Roussos, A. J. Marsh, S. Maglavera, "Enabling Pervasive Computing with Smart Phones" Pervasive Computing, IEEE, Volume 4, Issue 2, pp. 20-27, (2005).

[Saha, 03] D. Saha, A. Mukherjee, "Pervasive Computing: A Paradigm for the 21st Century", (2003).

[Salber, 99] D. Salber, A. Dey, and G. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications" In CHI, pp. 434–441, (1999).

[Schmidt, 02] A. Schmidt, “Ubiquitous Computing – Computing in Context”, PhD Thesis, Computing Department, Lancaster University, (2002).

[Schmidt, 06] A. Schmidt, S. Spiekermann, A. Gershman and F. Michahelles, “Real-World Challenges of Pervasive Computing”, Pervasive Computing, IEEE, Volume 5, Issue 3, pp. 91-13, (2006).

[Sheikh, 07] K. Sheikh, M. Wegdam, M. van Sinderen, “Middleware Support for Quality of Context in Pervasive Context-Aware Systems”, In proceedings of PerWare workshop, co-located with IEEE PerCom Conference, (2007).

[Shirazi, 04] B. Shirazi, M. Kumar, B. Y. Sung, “QoS Middleware Support for Pervasive Computing Applications” In Proceedings of the 37th Hawaii International Conference on System Sciences, (2004).

[Shyy, 00] D.J. Shyy, B. Rohani, “Indoor Location Technique for 2G and 3G Cellular/PC Networks” In Proceedings of the 25th IEEE Conference on Local Computer Networks, pp. 264–271, (2000).

[SNOMED CT, 07] http://www.snomed.org/snomedct/index.html, (2007).

[Soldatos, 07] J. Soldatos, I. Pandis, K. Stamatis, L. Polymenakos, J. L. Crowley, “Agent Based Middleware Infrastructure for Autonomous Context-Aware Ubiquitous Computing Services”, In Computer Communications, Volume 30, Issue 3. pp. 577-591, (2007).

[Stausberg, 03] J. Stausberg, D. Koch, J. Ingenerf, M. Betzler, "Comparing paper-based with Electronic Patient Records: Lessons Learned During a Study on Diagnosis and Procedure Codes.", Journal of the American Medical Informatics Association (JAmMedInformAssoc), pp. 470-477, (2003).

[Sun, 04] X.-H. Sun, A. R. Blatecky, "Middleware: The Key to Next Generation Computing" Preface for the Journal of Parallel and Distributed Computing, pp. 689–691, (2004).

[Trevisani, 04] E. Trevisani, A. Vitaletti, "Cell-ID Location Technique, Limits and Benefits: An Experimental Study," In Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA). IEEE Computer Society, pp. 51–60, (2004).

[Tyndall, 07] Tyndall National Institute
www.Tyndall.ie, (2007).

[UMLS, 07] http://www.openclinical.org/medTermUmls.html, (2007).

[van Bunningen, 05] A. H. van Bunningen, L. Feng, P. M.G. Apers, "Context for Ubiquitous Data Management", International Workshop on Ubiquitous Data Management (UDM), pp. 17-24, ISBN 0-7695-2411-7, (2005).

[Varshney, 03] U. Varshney "Pervasive Healthcare" IEEE Computer, Volume 36, Number 12, pp. 138-140, (2003).

[Varshney, 06] U. Varshney, "Managing Wireless Health Monitoring for Patients with Disabilities", IT Professional, Technology Solutions for Enterprise, Volume 8, Issue 6, pp. 12-16, (2006).

[Veríssimo, 02] P. Veríssimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday and J. Kaiser "CORTEX: Towards Supporting Autonomous and Cooperating Sentient Entities" In Proceedings of European Wireless conference, (2002).

[Wang, 04] X.H. Wang, T. Gu, D.Q. Zhang, H.K. Pung, "Ontology Based Context Modeling and Reasoning using OWL", Workshop on Context Modeling and Reasoning (CoMo-Rea) at PerCom, (2004).

[Weinstein, 02] E. Weinstein, P. Ho, B. Heisele, T. Poggio, K. Steele, and A. Agarwal, "Handheld Face Identification Technology in a Pervasive Computing Environment", In Proceedings of Pervasive Conference, (2002).

[Weiser, 91] M. Weiser, "The Computer for the 21st Century", In Proceedings of the Scientific American, Volume 265, Issue 3, pp. 94-104, (1991).

[WHO, 05] "Affordable Technology - Blood Pressure Measuring Devices for Low Resource Settings" World Health Organisation. WHO, (2005).

[WHO, 07] The World Health Organisation.
www.who.int, (2007).

[Winters, 03] J.M. Winters, Y. Wang, "Wearable Sensors and Telerehabilitation", Engineering in Medicine and Biology Magazine, IEEE Volume 22, Issue 3, pp. 56-65, (2003).

[Wooldridge, 97] M. Wooldridge, "Agent-Based Software Engineering", In Proceedings of IEE Journal of Software Engineering, Volume 144, pp. 26-37, (1997).

[WSMO, 07] Web Service Modelling Ontology
http://www.wsmo.org/, (2007).

[Zaslavsky, 04] A. Zaslavsky, "Mobile Agents: Can They Assist with Context Awareness?", IEEE International Conference on Mobile Data Management (MDM), pp. 304, (2004).

[Zhang, 06] L. Zhang and Z. Wang, "Integration of RFID into Wireless Sensor Networks: Architectures, Opportunities and Challenging Problems", (2006).

[Zhou, 05] H.Y. Zhou, K.M. Hou, J. Ponsonnaille, L. Gineste, C. de Vaulx, J. Coudon, G. De Sousa, J.J. Li, P. Chainais, R. Aufrère, A. Amamra and J.P. Chanet, "Real-Time Cardiac Arrhythmia Tele-Assistance and Monitoring Platform: RECATA", 3rd International Conference On Smart Homes and Health Telematic (ICOST), pp. 99-106, (2005).

APPENDIX A

Tyndall-Patient-Mote Code

Reading a Patient's Electrocardiogram (ECG) Signal

```
//**********************************************//
/*Chip type : ATmega128L
  Program type : Application
  Clock frequency : 4.000000 MHz
  Memory model : Small
  External SRAM size : 0
  Data Stack size : 1024*/
//**********************************************//

#include <iom128v.h>
#include <macros.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "global.h"   // include our global settings
#include "delay.c"
#include "adc.c"
#include "2401.c"
#include "adc_ADCH_ADCL.c"

unsigned char data[6];
char stop = 0;   //stop bit
int counter = 0;
int ADC4 = 0;

//********************************************************//
/*Initialisation of the Ports on the Atmega 128*/
//********************************************************//
void port_init(void)
{
 PORTA = 0xFF;
 DDRA  = 0x00;
 PORTB = 0xFF;
 DDRB  = 0x00;
 PORTC = 0xFF;   //m103 output only
 DDRC  = 0x00;
 PORTD = 0xFF;
 DDRD  = 0x00;
 PORTE = 0xFF;
 DDRE  = 0x00;
 PORTF = 0xFF;
 DDRF  = 0x00;
 PORTG = 0x1F;
 DDRG  = 0x00;
}

//#define START / STOP  (PORTA,0x00)
#define START_STOP   (PINA & (1<<0))
#define START  PORTA |= (1<<0)
#define STOP PORTA &= ~(1<<0)
```

```
#define STARTIO (DDRA,1)   //data direction
#define START_IN  DDRA &= ~(1<<0)
//#define VALVE  (PORTA,0x01)//
#define VALVE  (PORTA,0x01)
#define VALVE_CLOSE  PORTA |= (1<<1)
#define VALVE_OPEN  PORTA &= ~(1<<1)
#define VALVEIO (DDRA,1)   //data direction
#define VALVE_OUT  DDRA |= (1<<1)

//#define MOTOR  (PORTA,0x02)
#define MOTOR  (PORTA,0x02)
#define MOTOR_ON  PORTA |= (1<<2)
#define MOTOR_OFF  PORTA &= ~(1<<2)
#define MOTORIO (DDRA,2)   //data direction
#define MOTOR_OUT  DDRA |= (1<<2)

//*****************************************************//
//*Initialisation of the Tyndall-Patient-Mote board Ports
//*****************************************************//
void port_init_BPM(void)
{
 START_IN;
 STOP;   //disable pull up, on start stop i/p pin
 VALVE_OUT;
 VALVE_OPEN;
 MOTOR_OUT;
 MOTOR_OFF;
}
//*****************************************************//
/*Initialisation the ADC*/
//*****************************************************//
void adc_init(void)
{
 ADCSRA = 0x00; //disable adc
 ACSR  = 0x80;
 ADCSRA = 0x81;
}
//*****************************************************//
//Initialisation of the UART To a baud rate of
//9600/8bits/1 stop bit/no parity
//*****************************************************//
void uart0_init(void)
{
 UCSR0B = 0x00;   //disable while setting baud rate
 UCSR0A = 0x00;   //control and status register
 UCSR0C = 0x06;   //Asynchronous operation/
 UBRR0L = 0x19;   //set baud rate lo - 9600 bps
 UBRR0H = 0x00;   //set baud rate hi
 UCSR0B = 0x18;   //enables rx and tx
}
//*****************************************************//
//Comparator initialisation
//*****************************************************//
void comparator_init(void)
{
 ACSR = ACSR & 0xF7; //ensure interrupt is off before changing
 ACSR = 0x04;
}
```

```
//**********************************************************//
/*Initialise all peripherals*/
//**********************************************************//
void init_devices(void)
{
 //stop errant interrupts until set up
 CLI();   //disable all interrupts
 XDIV  = 0x00;  //xtal divider
 XMCRA = 0x00;   //external memory
 port_init();
 port_init_BPM();
 uart0_init();
 comparator_init();
 adc_init();

 MCUCR = 0x00;
 EICRA = 0x00;   //extended ext ints
 EICRB = 0x00;   //extended ext ints
 EIMSK = 0x00;
 TIMSK = 0x00;   //timer interrupt sources
 ETIMSK = 0x00;  //extended timer interrupt sources
 SEI();   //re-enable interrupts all peripherals are now initialised
}
//**********************************************************//
//Systolic + Diastolic Patient Readings
//**********************************************************//
void Get_DC_pressure_Data(char *data)
{
   int ADC4, ADC4H, ADC4L;
   ADC4 = ReadADC_ADCH_ADCL(0xC1);
   ADC4L = ADCL;
   ADC4H = ADCH;
   data++;
   data++;
   data++;
   *data = ADC4L;
   data++;
   *data = ADC4H;
}// End of Get_DC_pressure_Data
//**********************************************************//
//PULSE RATE
//**********************************************************//
void Get_AC_pressure_Data(char *data)
{
   int ADC4, ADC4H, ADC4L;
   ADC4 = ReadADC_ADCH_ADCL(0xC2);
   //adc0, ADLAR bit (ADMUX bit5)
   //cleared, ADC 10 bit result right justified
   //data must be read from ADCL first then
   //ADCH (use include file: adc_ADCH_ADCL.c)
   ADC4L = ADCL;
   ADC4H = ADCH;
   data++;
   data++;
   data++;
   *data = ADC4L;
   data++;
   *data = ADC4H;
}//End of Get_AC_pressure_Data
```

```
//**********************************************************//
//ECG Readings
//get analog ECG data from ECG patches (IA amp, adc0)
//**********************************************************//
void Get_ECG_Data(char *data)
{
   int ADC4, ADC4H, ADC4L;
   ADC4 = ReadADC_ADCH_ADCL(0xE0);
   ADC4L = ADCL;
   ADC4H = ADCH;
   data++;
   data++;
   data++;
   *data = ADC4L;
   data++;
   *data = ADC4H;
}//End of Get_ECG_Data

//**********************************************************//
/*Sends out data to COM Serial Poart*/
//**********************************************************//
void Serial_Data()
{
   putchar('A');      //Send out A
   putchar (data[3]); //Send out ADCL LOW (8 Bit)
   putchar('B');      //Send out B
   putchar (data[4]); //Send out ADCH High (8 Bit)
}// End of Serial_Data

void main(void)
{
   WDTCR = (1<<WDCE) | (1<<WDE);
   WDTCR = 0x00;
   init_devices();   //Initialise UART and ADC
   TESTPIN1_OUT;
   TESTPIN2_OUT;
   TESTPIN1_LO;
   TESTPIN2_LO;

   TESTPIN1_HI;       //Set LED PIN 1 ON for Debugging
   Delay_1ms(1000);  //Hardware Stabilisation delay

   //Set Data Arrays to Zero
   data[0] = 0;
   data[1] = 0;
   data[2] = 0;
      //ECG Reading
            while (TRUE)
            {
               TESTPIN2_HI;   //Set LED PIN 2 ON for Debugging
               Get_ECG_Data(data); //Returns continuous ECG signal
               Serial_Data(); //Send Data to Serial COM Port
               Delay_1ms(4);  //Delay ECG Reading Loop for 4 ms
               TESTPIN2_LO;   //Set LED PIN 2 OFF for Debugging

             }//End of While loop
}//End of Main
```

APPENDIX B

Tyndall-Patient-Mote ECG Readings

PL = = Power Device	**Interference (Power Supply)**		**Interference (Hand Tap)**	
Device and Technique	**Accuracy**	**BPM**	**Accuracy**	**BPM**
PL-Pulse-Visual	100%	68	100.00%	78
PL-Pulse -Static Area (25%)	0%	0	0.00%	0
PL-Pulse -Static Area (50%)	16.91%	11.5	30.13%	23.5
PL-Pulse -Static Area (75%)	95.59%	65	99.36%	77.5
PL-Pulse -Dynamic Area (10)	47.79%	32.5	32.69%	25.5
PL-Pulse -Dynamic Area (20)	54.41%	37	38.46%	30
PL-Pulse -Dynamic Area (30)	60.29%	41	43.59%	34
PL-Pulse -Static Slope (5)	99.26%	68	96.79%	80.5
PL-Pulse -Static Slope (10)	76.47%	52	79.49%	62
PL-Pulse -Static Slope (20)	27.21%	18.5	25.64%	20
PL-Pulse -Dynamic Slope (20)	99.26%	68	100%	80.5
PL-Pulse -Dynamic Slope (40)	99.26%	68	100.00%	80.5
PL-Pulse -Dynamic Slope (60)	99.26%	68	100.00%	80.5
PL-ECG-Visual	100%	68	100.00%	78
PL-ECG-Static Area (25%)	0.00%	0	4.49%	3.5
PL-ECG-Static Area (50%)	27.94%	19	33.97%	26.5
PL-ECG-Static Area (75%)	99.26%	68	96.79%	75.5
PL-ECG-Dynamic Area (10)	54.41%	37	51.28%	40
PL-ECG-Dynamic Area (20)	52.94%	36	53.21%	41.5
PL-ECG-Dynamic Area (30)	58.82%	40	55.77%	43.5
PL-ECG-Static Slope (5)	99.26%	68	100.00%	77.5
PL-ECG-Static Slope (10)	77.21%	52.5	76.92%	60
PL-ECG-Static Slope (20)	27.21%	18.5	25.64%	20
PL-ECG-Dynamic Slope (20)	99.26%	68	100%	77.5
PL-ECG-Dynamic Slope (40)	99.26%	68	100.00%	77.5
PL-ECG-Dynamic Slope (60)	99.26%	68	100.00%	77.5
Tyndall-Cuff-Visual	91%	59	90.83%	64
Tyndall-Cuff-Static Area (25%)	41.18%	108	97.12%	66.04
Tyndall-Cuff-Static Area (50%)	89.71%	75	75.47%	51.32
Tyndall-Cuff-Static Area (75%)	55.88%	98	74.37%	50.57
Tyndall-Cuff-Dynamic Area (10)	81.62%	55.5	41.06%	27.92
Tyndall-Cuff-Dynamic Area (20)	85.29%	58	43.84%	29.81
Tyndall-Cuff-Dynamic Area (30)	83.09%	56.5	41.26%	28.3
Tyndall-Cuff-Static Slope (5)	-71.93%	184.91	4.10%	133.21
Tyndall-Cuff-Static Slope (10)	84.35%	57.36	76.03%	51.7
Tyndall-Cuff-Static Slope (20)	73.81%	50.19	73.25%	49.81
Tyndall-Cuff-Dynamic Slope (20)	99.56%	68.3	77.9	83.02
Tyndall-Cuff-Dynamic Slope (40)	86.01%	58.49	89.34%	60.75
Tyndall-Cuff-Dynamic Slope (60)	88.79%	60.38	88.79%	60.38

	Resting		Running	
Device and Technique	**Accuracy**	**BPM**	**Accuracy**	**BPM**
PL-Pulse –Visual	100.00%	64	100.00%	88
PL-Pulse –Static Area (25%)	0.00%	0	0.00%	0
PL-Pulse –Static Area (50%)	26.56%	17	0.00%	0
PL-Pulse –Static Area (75%)	100.00%	64	96.59%	85
PL-Pulse –Dynamic Area (10)	52.34%	33.5	53.41%	47
PL-Pulse –Dynamic Area (20)	57.81%	37	56.82%	50
PL-Pulse –Dynamic Area (30)	63.28%	40.5	57.95%	51
PL-Pulse –Static Slope (5)	98.44%	65	100.00%	88
PL-Pulse –Static Slope (10)	86.72%	55.5	46.59%	41
PL-Pulse –Static Slope (20)	29.69%	19	26.14%	23
PL-Pulse –Dynamic Slope (20)	98.44	65	100.00%	88
PL-Pulse –Dynamic Slope (40)	98.44%	65	100.00%	88
PL-Pulse –Dynamic Slope (60)	98.44%	65	100.00%	88
PL-ECG-Visual	100.00%	64	100.00%	88
PL-ECG-Static Area (25%)	100.00%	64	0.00%	0
PL-ECG-Static Area (50%)	100.00%	64	2.27%	19
PL-ECG-Static Area (75%)	71.88%	46	62.50%	55
PL-ECG-Dynamic Area (10)	63.91%	34.5	36.36%	32
PL-ECG-Dynamic Area (20)	55.47%	35.5	37.50%	33
PL-ECG-Dynamic Area (30)	58.59%	37.5	36.36%	32
PL-ECG-Static Slope (5)	100.00%	64	97.73%	86
PL-ECG-Static Slope (10)	85.16%	54.5	42.05%	37
PL-ECG-Static Slope (20)	30.47%	19.5	22.73%	22.73
PL-ECG-Dynamic Slope (20)	100	64	97.73%	86
PL-ECG-Dynamic Slope (40)	100.00%	64	97.73%	86
PL-ECG-Dynamic Slope (60)	100.00%	64	97.73%	86
Tyndall-Cuff-Visual	100.00%	64	100%	88
Tyndall-Cuff-Static Area (25%)	82.81%	75	89.77%	97
Tyndall-Cuff-Static Area (50%)	98.44%	65	51.14%	131
Tyndall-Cuff-Static Area (75%)	98.44%	65	95.45%	84
Tyndall-Cuff-Dynamic Area (10)	43.75%	28	60.23%	53
Tyndall-Cuff-Dynamic Area (20)	40.62%	26	64.77%	57
Tyndall-Cuff-Dynamic Area (30)	42.19%	27	64.77%	57
Tyndall-Cuff-Static Slope (5)	-40.27%	153.77	-65.91%	234
Tyndall-Cuff-Static Slope (10)	100.00%	64	69.32%	115
Tyndall-Cuff-Static Slope (20)	100.00%	64	95.45%	84
Tyndall-Cuff-Dynamic Slope (20)	100	64	98.86%	89
Tyndall-Cuff-Dynamic Slope (40)	100.00%	64	97.73%	90
Tyndall-Cuff-Dynamic Slope (60)	100.00%	64	98.86%	89

	Running to Resting		Cuff A	
Device and Technique	**Accuracy**	**BPM**	**Accuracy**	**BPM**
PL-Pulse-Visual	100.00%	87	100.00%	75
PL-Pulse -Static Area (25%)	0.00%	0	0.00%	0
PL-Pulse -Static Area (50%)	11.49%	10	1.33%	1
PL-Pulse -Static Area (75%)	97.70%	85	96.00%	72
PL-Pulse -Dynamic Area (10)	58.62%	51	61.33%	46
PL-Pulse -Dynamic Area (20)	59.77%	52	65.33%	49
PL-Pulse -Dynamic Area (30)	62.07%	54	66.67%	50
PL-Pulse -Static Slope (5)	100.00%	87	100%	75
PL-Pulse -Static Slope (10)	50.57%	44	61.33%	46
PL-Pulse -Static Slope (20)	29.89%	26	25.33%	19
PL-Pulse -Dynamic Slope (20)	100.00%	87	100.00%	75
PL-Pulse -Dynamic Slope (40)	100.00%	87	100.00%	75
PL-Pulse -Dynamic Slope (60)	100.00%	87	100.00%	75
PL-ECG-Visual	100.00%	87	100.00%	75
PL-ECG-Static Area (25%)	0.00%	0	0.00%	0
PL-ECG-Static Area (50%)	0.00%	0	33.33%	25
PL-ECG-Static Area (75%)	89.66%	78	98.67%	74
PL-ECG-Dynamic Area (10)	55.17%	48	69.33%	52
PL-ECG-Dynamic Area (20)	58.62%	51	72.00%	54
PL-ECG-Dynamic Area (30)	58.62%	51	73.33%	55
PL-ECG-Static Slope (5)	100.00%	87	100.00%	75
PL-ECG-Static Slope (10)	41.38%	36	61.33%	46
PL-ECG-Static Slope (20)	26.44%	23	21.33%	16
PL-ECG-Dynamic Slope (20)	100.00%	87	100.00%	75
PL-ECG-Dynamic Slope (40)	100.00%	87	100.00%	75
PL-ECG-Dynamic Slope (60)	100.00%	87	100.00%	75
Tyndall-Cuff-Visual	100.00%	87	98.30%	73
Tyndall-Cuff-Static Area (25%)	74.71%	109	94.52%	77
Tyndall-Cuff-Static Area (50%)	50.57%	130	98.63%	74
Tyndall-Cuff-Static Area (75%)	96.55%	84	98.63%	74
Tyndall-Cuff-Dynamic Area (10)	97.70%	85	90.41%	66
Tyndall-Cuff-Dynamic Area (20)	100.00%	87	83.56%	61
Tyndall-Cuff-Dynamic Area (30)	98.85%	88	76.71%	56
Tyndall-Cuff-Static Slope (5)	-56.23%	223	-49.32%	182
Tyndall-Cuff-Static Slope (10)	78.16%	106	100.00%	73
Tyndall-Cuff-Static Slope (20)	98.85%	86	100.00%	73
Tyndall-Cuff-Dynamic Slope (20)	97.70%	89	100.00%	73
Tyndall-Cuff-Dynamic Slope (40)	97.70%	89	100.00%	73
Tyndall-Cuff-Dynamic Slope (60)	97.70%	89	100.00%	73

	Cuff AH		Cuff B150	
Device and Technique	**Accuracy**	**BPM**	**Accuracy**	**BPM**
PL-Pulse –Visual	100.00%	75	100.00%	73
PL-Pulse –Static Area (25%)	0.00%	0	0.00%	0
PL-Pulse –Static Area (50%)	0.00%	0	2.74%	2
PL-Pulse –Static Area (75%)	93.33%	70	93.15%	68
PL-Pulse –Dynamic Area (10)	64.33%	46	65.75%	48
PL-Pulse –Dynamic Area (20)	62.67%	47	68.49%	50
PL-Pulse –Dynamic Area (30)	65.33%	49	72.60%	53
PL-Pulse –Static Slope (5)	100.00%	75	100.00%	73
PL-Pulse –Static Slope (10)	61.33%	46	64.38%	47
PL-Pulse –Static Slope (20)	24.00%	18	26.03%	19
PL-Pulse –Dynamic Slope (20)	100.00%	75	100.00%	73
PL-Pulse –Dynamic Slope (40)	100.00%	75	100.00%	73
PL-Pulse –Dynamic Slope (60)	100.00%	75	100.00%	73
PL-ECG-Visual	100.00%	75	100.00%	73
PL-ECG-Static Area (25%)	0.00%	0	0.00%	0
PL-ECG-Static Area (50%)	26.67%	20	28.77%	21
PL-ECG-Static Area (75%)	100.00%	75	100.00%	73
PL-ECG-Dynamic Area (10)	70.67%	53	72.60%	53
PL-ECG-Dynamic Area (20)	73.33%	55	73.97%	54
PL-ECG-Dynamic Area (30)	76.00%	57	73.97%	54
PL-ECG-Static Slope (5)	100.00%	75	100.00%	73
PL-ECG-Static Slope (10)	61.33%	46	64.38%	47
PL-ECG-Static Slope (20)	22.67%	17	24.66%	18
PL-ECG-Dynamic Slope (20)	100.00%	75	100.00%	73
PL-ECG-Dynamic Slope (40)	100.00%	75	100.00%	73
PL-ECG-Dynamic Slope (60)	100.00%	75	100.00%	73
Tyndall-Cuff-Visual	98.30%	73	99.70%	72
Tyndall-Cuff-Static Area (25%)	68.49%	96	43.84%	114
Tyndall-Cuff-Static Area (50%)	97.26%	75	100.00%	73
Tyndall-Cuff-Static Area (75%)	97.26%	75	100.00%	73
Tyndall-Cuff-Dynamic Area (10)	87.67%	64	89.04%	65
Tyndall-Cuff-Dynamic Area (20)	84.93%	62	82.19%	60
Tyndall-Cuff-Dynamic Area (30)	80.82%	59	78.08%	57
Tyndall-Cuff-Static Slope (5)	-15.07%	157	-16.44%	158
Tyndall-Cuff-Static Slope (10)	98.63%	74	97.26%	71
Tyndall-Cuff-Static Slope (20)	98.63%	74	98.36%	72
Tyndall-Cuff-Dynamic Slope (20)	98.63%	74	97.26%	71
Tyndall-Cuff-Dynamic Slope (40)	98.63%	74	97.26%	71
Tyndall-Cuff-Dynamic Slope (60)	98.63%	74	97.26%	71

	Cuff C	
Device and Technique	**Accuracy**	**BPM**
PL-Pulse-Visual	100.00%	68
PL-Pulse -Static Area (25%)	0.00%	0
PL-Pulse -Static Area (50%)	5.88%	4
PL-Pulse -Static Area (75%)	94.12%	64
PL-Pulse -Dynamic Area (10)	67.65%	46
PL-Pulse -Dynamic Area (20)	70.59%	48
PL-Pulse -Dynamic Area (30)	73.53%	50
PL-Pulse -Static Slope (5)	100.00%	68
PL-Pulse -Static Slope (10)	76.47%	52
PL-Pulse -Static Slope (20)	25.00%	17
PL-Pulse -Dynamic Slope (20)	100.00%	68
PL-Pulse -Dynamic Slope (40)	100.00%	68
PL-Pulse -Dynamic Slope (60)	100.00%	68
PL-ECG-Visual	100.00%	68
PL-ECG-Static Area (25%)	0.00%	0
PL-ECG-Static Area (50%)	20.59%	14
PL-ECG-Static Area (75%)	100.00%	68
PL-ECG-Dynamic Area (10)	75.00%	51
PL-ECG-Dynamic Area (20)	75.00%	51
PL-ECG-Dynamic Area (30)	76.47%	52
PL-ECG-Static Slope (5)	100.00%	68
PL-ECG-Static Slope (10)	75.00%	51
PL-ECG-Static Slope (20)	22.06%	15
PL-ECG-Dynamic Slope (20)	100.00%	68
PL-ECG-Dynamic Slope (40)	100.00%	68
PL-ECG-Dynamic Slope (60)	100.00%	68
Tyndall-Cuff-Visual	100.00%	68
Tyndall-Cuff-Static Area (25%)	26.47%	118
Tyndall-Cuff-Static Area (50%)	100.00%	68
Tyndall-Cuff-Static Area (75%)	100.00%	68
Tyndall-Cuff-Dynamic Area (10)	97.06%	66
Tyndall-Cuff-Dynamic Area (20)	98.53%	67
Tyndall-Cuff-Dynamic Area (30)	98.53%	67
Tyndall-Cuff-Static Slope (5)	-16.18%	147
Tyndall-Cuff-Static Slope (10)	100.00%	68
Tyndall-Cuff-Static Slope (20)	98.53%	67
Tyndall-Cuff-Dynamic Slope (20)	100.00%	68
Tyndall-Cuff-Dynamic Slope (40)	94.12%	72
Tyndall-Cuff-Dynamic Slope (60)	92.65%	73

APPENDIX C

DMS-JAI GUI

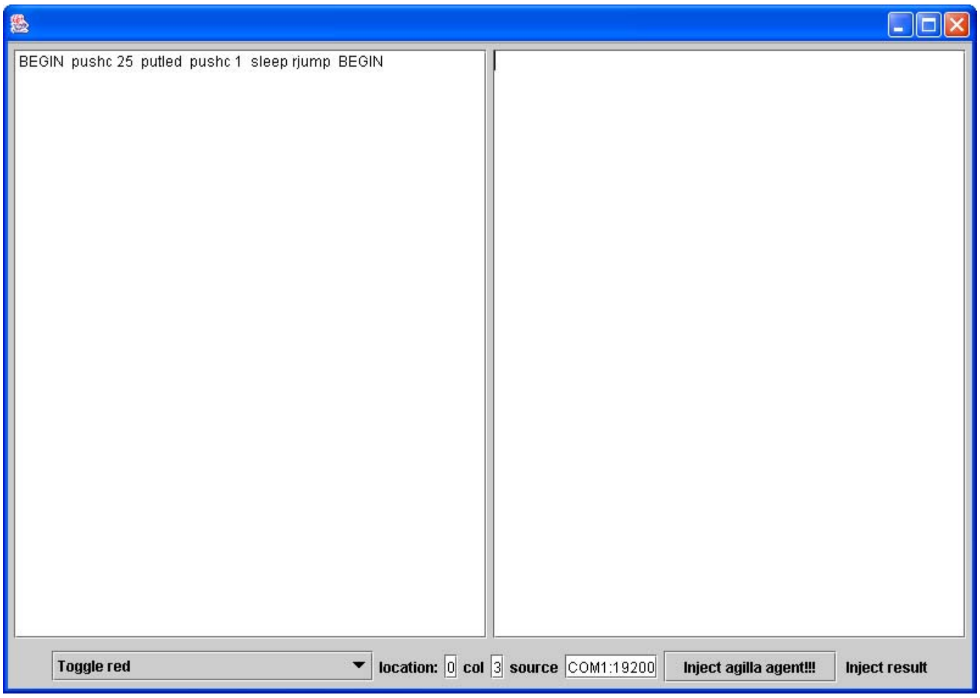


Figure C.1: DMS-JAI Client ready to inject Agilla Source Code.

```
//Sample Jade Server Code. The DMSJAIServerAgent receives Agilla code from the DMS-JAI
//Client (cf. figure C.1) and injects it into the WSN.
//DMSJAIServerAgent.java

package agents;

import java.util.Iterator;
import java.util.Vector;
import edu.wustl.mobilab.agilla.*;
import edu.wustl.mobilab.agilla.messages.*;
import edu.wustl.mobilab.agilla.variables.AgillaLocation;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.ParallelBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.lang.acl.ACLMessage;
import jade.core.AID;

public class DMSJAIServerAgent extends Agent
{
		private Vector manageQueue = new Vector();
		private Vector msgQueue = new Vector();
		private Vector injectQueue = new Vector();
```

```
	public static int injectResult = 0 ;
	private static Vector receiveAgillaData = new Vector();
	private static Vector  jadeAgillaDataQueue = new Vector();
	private boolean sentTrue = true;

	AgentInjector injector;

	class ReceiveMsgCyclicBehaviour extends CyclicBehaviour
	{
		private String myStep;
		private int num = 0;

		public ReceiveMsgCyclicBehaviour(Agent a, String step)
		{
			super(a);
			myStep = step;
			System.out.println("ReceiveMsgCyclicBehaviour!");
		}
		public void action(){

		ACLMessage codeMsg = receive();
		System.out.println("ReceiveMsgCyclicBehaviour" + (this.num++));
		if(codeMsg != null)
		{
			switch(codeMsg.getPerformative())
			{
				case(ACLMessage.SUBSCRIBE):
					ACLMessage agree = codeMsg.createReply();
					agree.setPerformative(ACLMessage.AGREE);
					agree.setContent("A");
					System.out.println("AGREE");
					send(agree);
					addMsg(codeMsg);    // add inject code
					return;

				case(ACLMessage.REQUEST):
					String dataTime = null;
					dataTime = codeMsg.getContent();
					System.out.println(dataTime);
					int time = 1000;
					time = Integer.parseInt(dataTime);
					if ((time < 60000000) && (time > 1))
					{
						//Do Nothing

					}
					else
					{
						System.out.println("error");
					}
					return;

				case(ACLMessage.CANCEL):
					return;

			}//End of Switch
		}//End of IF
		block();
	}//End of Class ReceiveMsgCyclicBehaviour
}//End of Class DMSJAIServerAgent
```

APPENDIX D

Sensor Validation Results

D.1 Area Technique

The performance results for the Static and Dynamic Area techniques are given. They are evaluated under noisy conditions (i.e. electrical and physical interference).

D.1.1 Static Area Technique

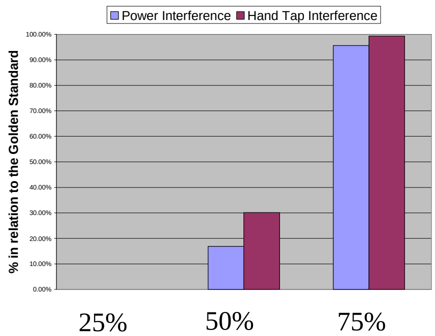


Figure D.1 (A): Power Lab Pulse with a Static Area technique applied.

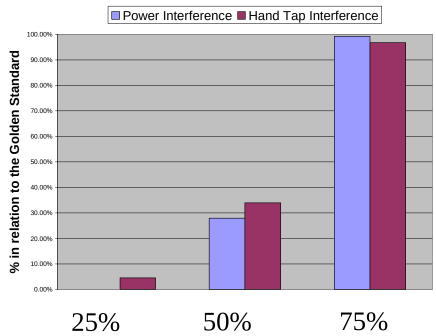


Figure D.1 (B): Power Lab ECG with a Static Area technique applied.

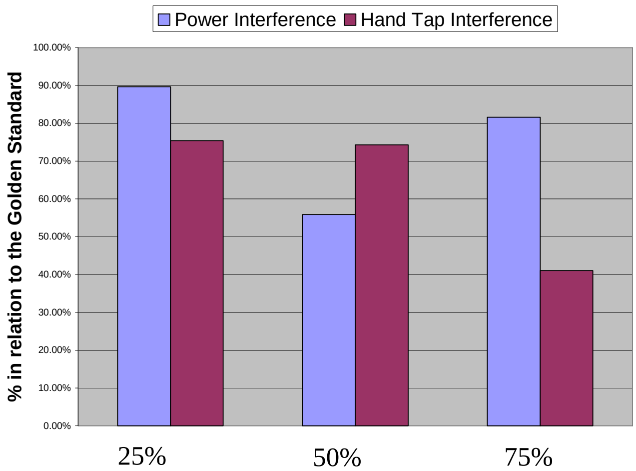


Figure D.1 (C): Tyndall-DMS-Mote ECG with Static Area technique applied.

The Power Lab Pulse and ECG produced excellent results of 95% plus, once the appropriate percentage of 75% was applied to these specific graphs (cf. figure D.1 (A), (B)). Conversely the Tyndall-DMS-Mote ECG readings degraded as the percentage parameter increased (cf. figure D.1 (C)). This may be attributed to the noisy condition of the original signal (i.e. average technique could not remove all of the interference). The static area technique may only operate effectively under very strict conditions. The form of the relevant graphs needs to be visually examined to calculate the appropriate "sample area". This requires very time consuming manual calculations.

D.1.2 Dynamic Area Technique

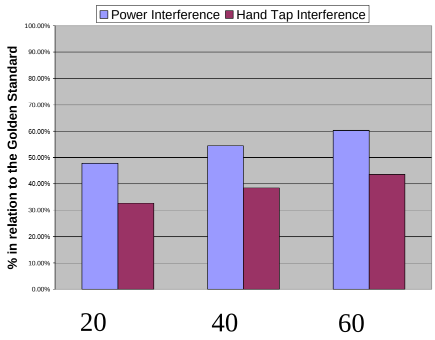


Figure D.2: (A) Power Lab Pulse, with Dynamic Area technique applied.

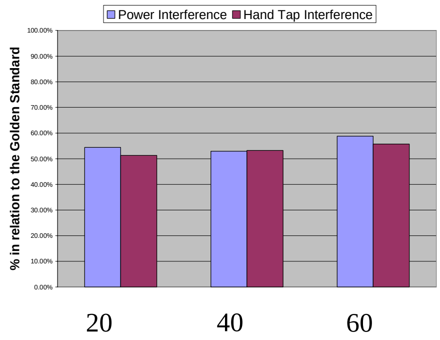


Figure D.2: (B) Power Lab ECG, with Dynamic Area technique applied.

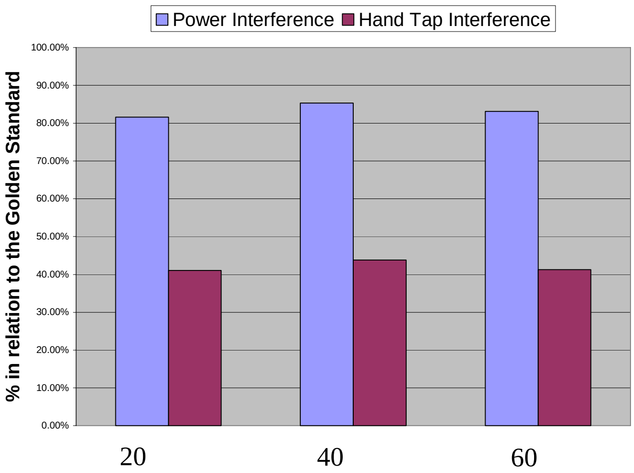


Figure D.2 (C): Tyndall-DMS-Mote ECG, with Dynamic Area technique applied.

The dynamic area technique was examined under similar conditions to the static area technique (i.e. averaging of 10 points). The Power Lab Pulse and ECG signals achieved a poor average of approximately 50%. Only a slight improvement of 80% is achieved for the Tyndall-DMS-Mote ECG power interference readings.

D.2 Slope Technique

The performance results for the Static and Dynamic Slope techniques are given. They are evaluated under noisy conditions (i.e. power and hand tap interference). The static slope uses an averaging technique of 5, 10 and 20 points. The dynamic slope technique alters its minimal ECG R-R interval based on the R-R intervals of the previous 20, 40 and 60 cycles.

D.2.1 Static Slope Technique

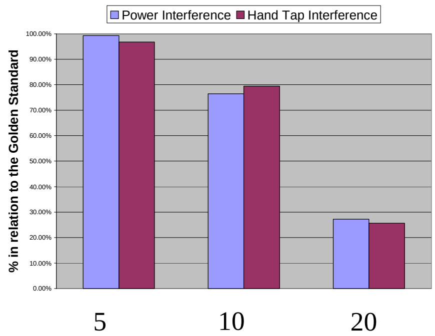


Figure D.3 (A): Power Lab Pulse, with Static Slope technique applied.

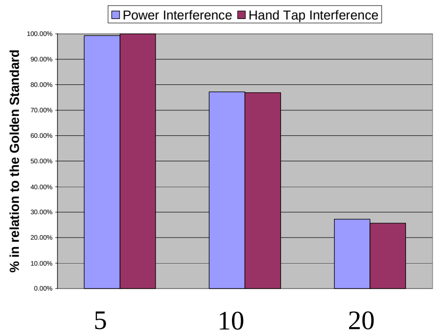


Figure D.3 (B): Power Lab ECG, with Static Slope technique applied.

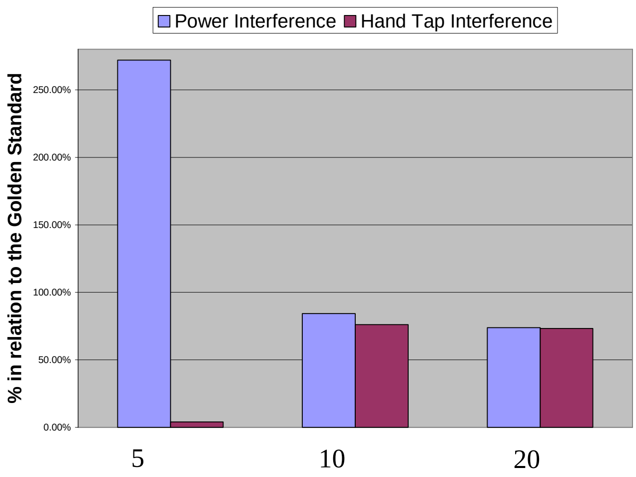


Figure D.3 (C): Tyndall ECG, with Static Slope technique applied.

In contrast to the static Area technique, an increase of the average points decreases the level of accuracy (cf. figure D.3 (A), (B)). This may be attributed to the over averaging of signal points (cf. figure 4.6 (D) chapter 4). The results generated for the Tyndall-DMS-Mote ECG power interference with an average of 5 points are extremely poor. This is a result of the slope technique picking up too many of the smaller interfering signals. Under the right conditions (appropriate averaging technique) the slope technique may achieve an accuracy level of 90% plus.

D.2.2 Dynamic Slope Technique

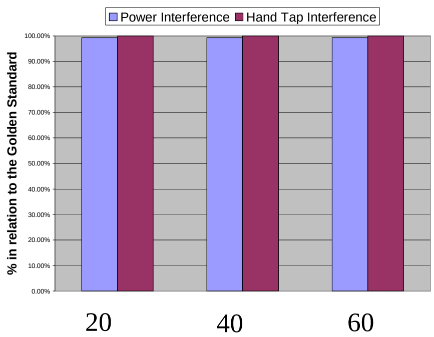


Figure D.4 (A): Power Lab Pulse, with Dynamic Slope technique applied.

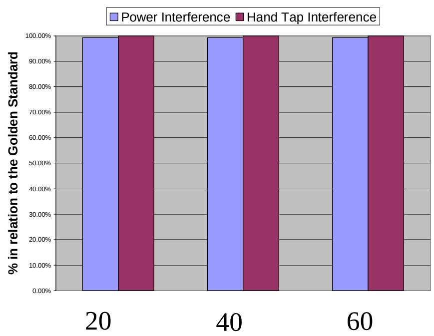


Figure D.4 (B): Power Lab ECG, with Dynamic Slope technique applied.

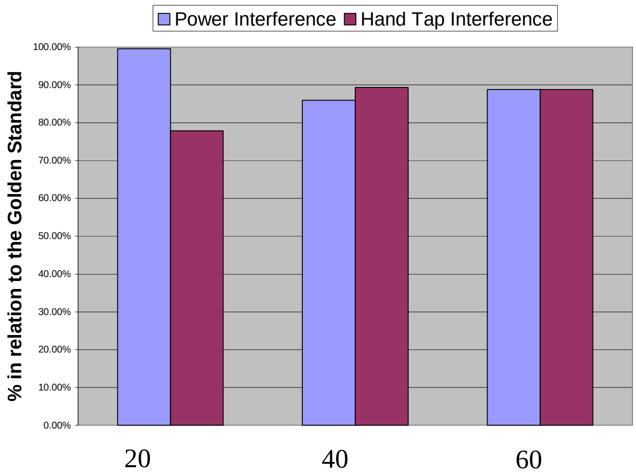


Figure D.4 (C): Tyndall-DMS-Mote ECG, with Dynamic Slope technique applied.

The Dynamic Slope averages the number of valid beats over time for the previous 20, 40 and 60 cycles. This approach enables the Dynamic Slope technique to allow for a patient's increase or decrease in heart rate (fluctuation in their ECG R-R interval). Regarding the Power Lab Pulse and ECG a percentage rate of 98% plus is achieved (cf. figure D.4 (A), (B)). This is extremely successful considering the level of interference involved. The Tyndall-DMS-Mote ECG produced an average of 88% which is relatively successful considering the limited resources in relation to the Power Lab unit (cf. figure D.4 (C)).