Data Management within mHealth Environments: Patient Sensors, Mobile Devices, and Databases

JOHN O'DONOGHUE, Health Information Systems Research Centre, Ireland
JOHN HERBERT, University College Cork, Ireland

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 be a large data source. 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 article presents the Data Management System (DMS) architecture. It is designed to deliver quality 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. Two of the DMS components are presented: (1) data validation and (2) data consistency. The DMS components have been rigorously evaluated using various medical-based test cases.

This article 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.

Categories and Subject Descriptors: E.0 [**Data**]: General

General Terms: Management, Measurement

Additional Key Words and Phrases: Data quality, data management, mHealth, body area network

ACM Reference Format:
O'Donoghue, J. and Herbert, J. 2012. Data management within mHealth environments: Patient sensors, mobile devices, and databases. ACM J. Data Inform. Quality 4, 1, Article 5 (October 2012), 20 pages.
DOI = 10.1145/2378016.2378021 http://doi.acm.org/10.1145/2378016.2378021

5

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 1991]

The role of pervasive computing brings with it new opportunities to assist us in our daily lives. It enables us to gather knowledge and interact with our world environment

Authors' addresses: J. O'Donoghue, Health Information Systems Research Centre, University College Cork, Ireland; email: john.odonoghue@ucc.ie; J. Herbert, Department of Computer Science, University College Cork, Ireland; email: j.herbert@cs.ucc.ie.

DOI 10.1145/2378016.2378021 http://doi.acm.org/10.1145/2378016.2378021

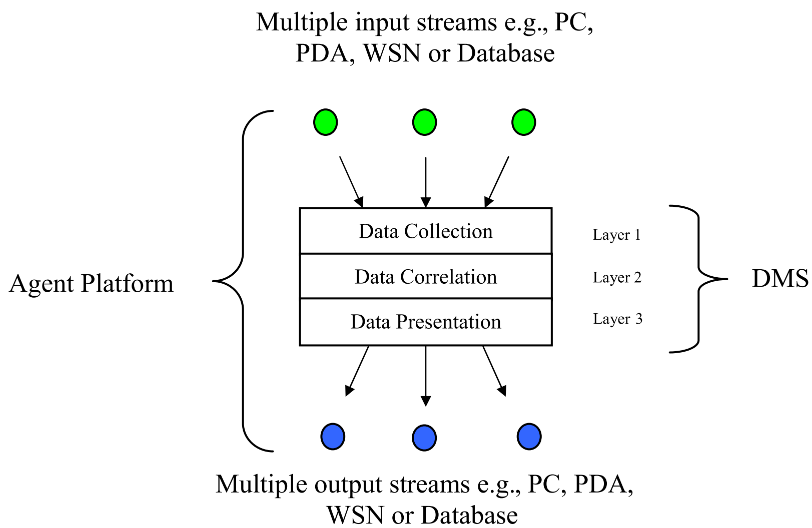


Fig. 1. DMS architecture.

in a seamless real time fashion. 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 [Ballagas et al. 2006; Roussos et al. 2005]. 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 [Drugge et al. 2006; Lorincz et al. 2004; Lubrin et al. 2005]. 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 utilized these pervasive tools.

An intelligent data management infrastructure is needed to realize the potential of these data-rich resources. This will enable 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 and Herbert 2005]. 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.

The current DMS prototype is designed to manage the complex dynamic datasets within a medical environment. Presented in Figure 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 requirements 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.

1.1 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. Software agents are ideally suited to function within a context-rich environment, due to their intelligent built-in autonomous reasoning. The potential improvement in providing effective patient service is increased. 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 high quality 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 2004]. Agents are designed to operate effectively within a context-rich environment. This article 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.

1.2 Contributions

The primary contribution of this research is the Data Management System (DMS). It is designed to advance the development of context-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 the following.

— Sensor validation;
— Data consistency.

In collaboration with the Tyndall National Institute and the Department of Medicine (UCC), the Tyndall-DMS-Mote [O'Flynn et al. 2006, 2007] 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.

1.3 Article Structure

This article includes the following sections.

— Background research and related work within the pervasive community is presented in Section 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. [Angrove et al. 2007; Barton et al. 2005; O'Donoghue and Herbert 2005, 2006f; O'Flynn et al. 2006, 2007].
— In Section 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 Section 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 et al. 2006a].
— In Section 5, 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.

2. BACKGROUND 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 nonexistent.

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 [Chagoyen and Pascual-Montano 2004]. 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 section 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 organize 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 oriented medical-based pervasive data management infrastructure.

2.2 Pervasive Data

In this section, a key medical data source, a wireless patient monitoring device, is assessed, along with the handheld devices that utilize 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 et al. 2003]. This approach results in poor patient care, while increasing costs and medical errors.

Pervasive patient care is one approach that may improve or cooperate with the paper-based approach [Geer 2006a, 2006b]. Numerous wireless patient sensing devices have been developed to assist in providing a higher quality of service within the medical environment [Barton et al. 2005; Fensli et al. 2005; O'Flynn et al. 2006; Winters and Wang 2003]. However data generated by a patient sensing device is not always one hundred percent accurate or of a high quality. Data may be corrupted due to power/radio interference, poor coding, faulty sensors, or poor sensor contact.

In Section four of this article 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 offline 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 mHealth Devices within a Healthcare Environment. The potential role that pervasive computing mobile devices can play within a medical environment is evident [Banitasa et al. 2004; Kearney et al. 2006; Weinstein et al. 2002]. A major focus of this article 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 article 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.

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 and Mukherjee 2003]. 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 et al. 2006]. Middleware technologies are the key to next generation computing [Nornam 1998; Sun and Blatecky 2004]. Middleware can utilize 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 that 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 current majority of sensing computing systems are reactive in nature. By introducing proactiveness 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 that may annoy or mislead the user [Saha and Mukherjee 2003].

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 regarding 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. Effective middleware decouples applications from the underlying heterogeneous sensors, enabling end-users to concentrate solely on relevant datasets [Kavimandan et al. 2006; Sheikh et al. 2007].

Pervasive computing technologies are developed to provide real-time unobtrusive user services in dynamic heterogeneous environments [Bardram 2004; Bardram and Christensen 2007; Shirazi et al. 2004]. General purpose distributed middleware such as RMI, CORBA, or DCOM is traditionally used to provide the necessary communication infrastructure between multiple distributed objects. For 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 than 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 utilized by the distributed components within the DMS presented in this article. In the context of software engineering, an agent can be defined as the following.

> "An entity within a computer system environment that is capable of flexible, autonomous actions with the aim of complying with its design objectives." [Woolridge 1997]

A pervasive environment is complex in nature; it requires sophisticated communication and reasoning infrastructures to meet the end user's run-time requirements. Agent-based middleware contains sufficient autonomy, reactivity, proactiveness, and social ability to facilitate the development of context-based applications [Bisdikian et al. 2002; Woolridge 1997; Zaslavsky 2004]. 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 2001]. An agent middleware manages the coordination, cooperation, and interoperability of distributed components by linking applications with their underlying low level software and hardware infrastructures. 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 [Soldtos et al. 2007]. Most agent-based middleware is designed to communicate, migrate, and have a certain degree of built-in autonomy. This enables it to trigger events with little or no intrusion on the end-user (specifically routine events). Agent middleware can be extremely effective when combined with specialized hardware or software elements. To fully understand the context of any environment, sensory data needs to be obtained and adequately processed (cognitively understood). By merging intelligent middleware with specialized sensory data analysis tools, relevant context datasets may be collected, correlated, and disseminated to interested parties within the network. As middleware in general operates at a very high logical level, the use of lower level applications/tools maximizes 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 and Georgeff 1991]. The use of agents within a healthcare environment has been shown to increase productivity in assisting medical staff during their daily tasks. Mazzi et al. [2001] 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 that may assist the medical practitioner; for example, the following.

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 et al. [2001] presented a number of scenarios where software agents may be broken down into logical data analyzing tools to assist the medical staff in providing 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.

2.4 A Data\Information Quality-Based Environment

Smart mobile devices, embedded sensors and high bandwidth networking infrastructures are assisting the pervasive paradigm. 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 to the end user at the correct moment. 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 infrastructure 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 recognized. Such information can then be passed on to the end-user where a final acceptance or rejection of the sent data can be processed [Khoukhi and Cherkaoui 2005; Varshney 2006]

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 degraded service and potential bottlenecks [Choi et al. 2004]. 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 et al. [2004] present PICO (Pervasive Information Community Organization) middleware. The PICO middleware provides 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 and Hochstatter 2005]

Distributed middleware provides the necessary services that enable front end and back end devices/applications to interact as a single entity. These may include DCOM, CORBA, RMI, or agent-based middleware Aglets, Jade, and Agilla. [Kisazumi et al. 2003] presents CAMPUS context-aware middleware to construct context-aware applications. The CAMPUS data-gathering approach has 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 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 et al. 2003; Huebscher and McCann 2004] presents adaptive middleware for context-aware applications. Here the location of the mobile user is matched against all known real-world devices that are of interest.

2.4.3 Quality of Data\Information. 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 et al. 2000] outlined 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, but QoD is not. In relation to this article, the term QoD is defined as providing the end user with value-added data that 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 high quality service within such an environment, all QoS, QoC, and QoD parameters need to be satisfied.

A large variety of distributed middleware exists. 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 program is used for its ability to intelligently filter nonrelevant real world events and to seamlessly interact with end-users in a nonintrusive manner. The agent platform also provides a mature computing paradigm. It contains an inherently sentient design philosophy that directs the software developer to focus on key user context and data quality requirements.

3. THE DATA MANAGEMENT SYSTEM (DMS) ARCHITECTURE

Pervasive environments generate large quantities of data. If utilized 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 and Herbert 2005, 2006f]. 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 article, is the Data Management System (cf. Figure 1). It is built on agent middleware and supports context-based data management within a pervasive environment. Outlined in Figure 2, 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) back end servers, where processing and communication are less of an issue (cf. Figure 2). The back end 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 back end servers. This provides context-based communication, enabling data to be transmitted in a coordinated and timely manner. Agent middleware (Jade) is utilized to provide the necessary communication and reasoning capabilities between the front end and back end devices.

Figure 1 is an overview of the DMS architecture. It contains three fundamental data management layers designed to deliver 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.

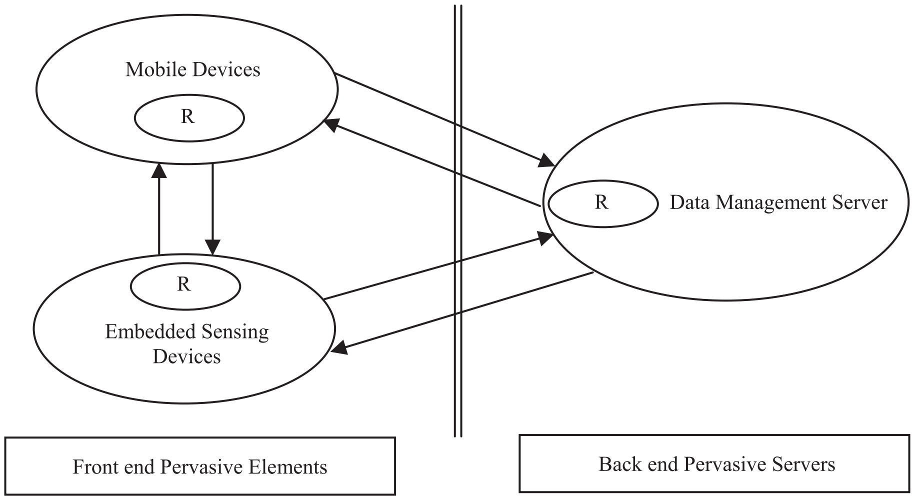


Fig. 2. Context-based pervasive data management. R is local inbuilt reasoning capability.

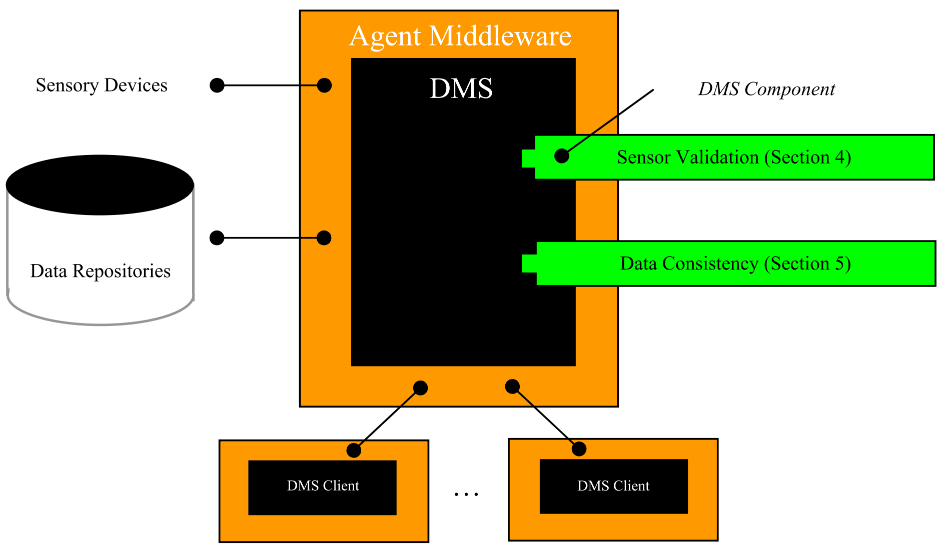


Fig. 3. A logical overview of the DMS architecture and DMS components. Mobile DMS clients may interact with the DMS server. Data pools include sensory devices and large data storage facilities.

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.

Two DMS components are created and provide the primary functionality of the DMS architecture (cf. Figure 3). The DMS presented in this article is healthcare oriented. The components developed here may be carried over to other application domains, such as banking or factory automation.

(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.

(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 synchronized with all known data sources (front end and back end). This approach helps to reduce information overload.

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 two DMS component prototypes presented in this article are designed to target real-world data management medical scenarios.

4. SENSOR VALIDATION WITHIN A PERVASIVE MEDICAL ENVIRONMENT

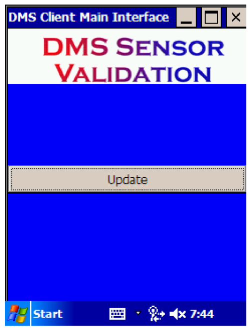
4.1 Introduction

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 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 that a high QoS is provided. Presented, is a Data Management System-Validation Model (DMS-VM) [O'Donoghue et al. 2006a], 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 analyze the sensory datasets to verify their integrity. This novel approach provides a higher QoS within a context-aware medical environment.

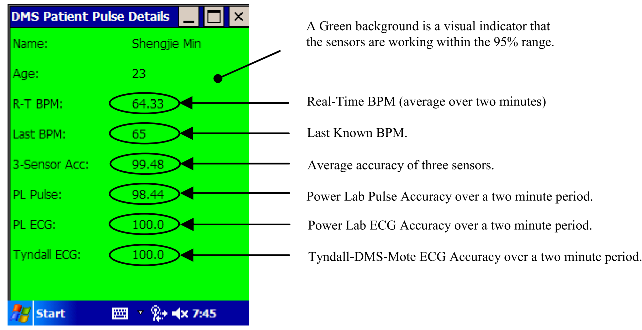
One of the core design features of the DMS (Data Management System) [O'Donoghue and Herbert 2006f] is to provide a higher QoS within a pervasive environment. The DMS-VM includes two validation protocols to ensure medical practitioners receive accurate patient datasets. The first protocol is "isolated 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(a); against a pulse sensor). The loss of transmitted signals within the medical community is unacceptable. Golmie et al. [2005] 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. A detailed description of the DMS-VM development and test care environment is presented in O'Donoghue et al. [2006a].

4.2 Transmission of Validated Physiological Signals

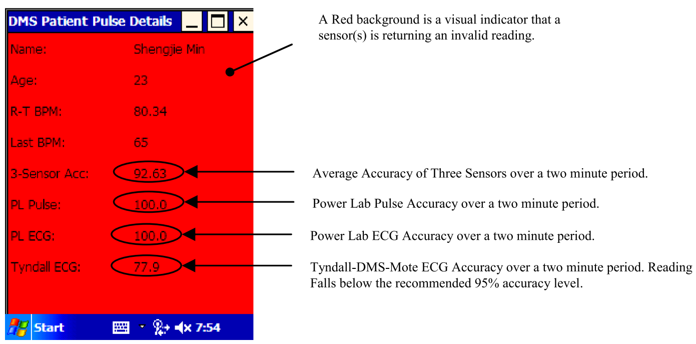
It is important to identify as quickly as possible if a sensor is not operating within its tolerance levels. Reacting to a sensor failure in real time can greatly enhance the overall reliability and dependability. Within a pervasive environment, sensor validation can greatly enhance the quality of data delivered to the end-user. Figures 4(a) and



(a) Initial DMS-VM screen.



(b) Valid patient pulse details with three pulse reading sources.



(c) Invalid patient pulse details with three pulse reading sources.

Fig. 4.

4(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(b) represent valid BPM readings. The readings are 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(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.

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 analyzed 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.

4.3 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 is 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 home.

5. MANAGING DATA CONSISTENCY WITHIN AN MHEALTH DISTRIBUTED ENVIRONMENT

5.1 Introduction

Presented in this section, is the Data Management System-Data Consistency Model (DMS-DCM) component. It is designed to ensure that vital datasets are synchronized between front end devices and back end 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 synchronized in a timely manner.

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 to them at the patient point-of-care. 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 et al. 2006b]. It is designed to intelligently interact with servers, databases, 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 information 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 employs two key operations, read and write. In relation to data consistency a write operation cannot be executed in isolation. It needs to verify that no other user is interacting with the current dataset and that correct datasets are replicated among all users. This is referred to as strong data consistency [Pitoura and Bhargava 1999]. The DMS-DCM applies an approach similar to the view-based consistency of Huang et al. [2001]. 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. Multicast-based middleware is presented in Chrysanthis et al. [2003]. 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 et al. [2001], 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.

5.3 Test Case Environment

All experiments are conducted on patient pulse rate readings in an offline mode. The pulse readings are sampled over a four-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.

5.4 Data Priority

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. In this article one data consistency management scenario is evaluated.

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.

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 is presented (cf. Table I) to manage this data consistency requirement.

This new set of data consistency rules was applied within a simulated environment with three patient levels of activity: resting, mildly active, and active. The results are presented in Figure 5 and Table II.

The gradual data consistency rules have prioritized the critical data and transmitted accordingly. The overhead associated with high-risk patients can be seen with a

Table I. Gradual Data Consistency Rules

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

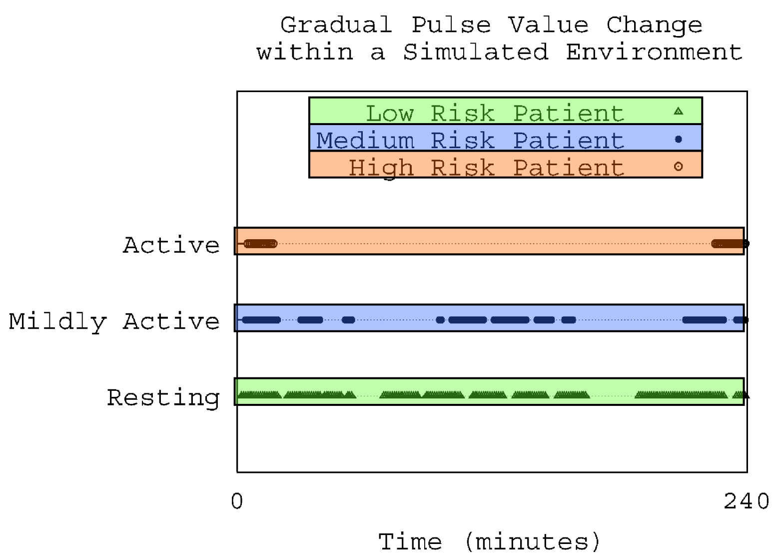


Fig. 5. Gradual sensor reading change under various patient states.

Table II. Gradual Patient Updates under Various Patient States

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

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.

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 6, are the results based on the data consistency rules (cf. Table III) 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 on to the medical practitioner. This resulted in 60% inconsistency.

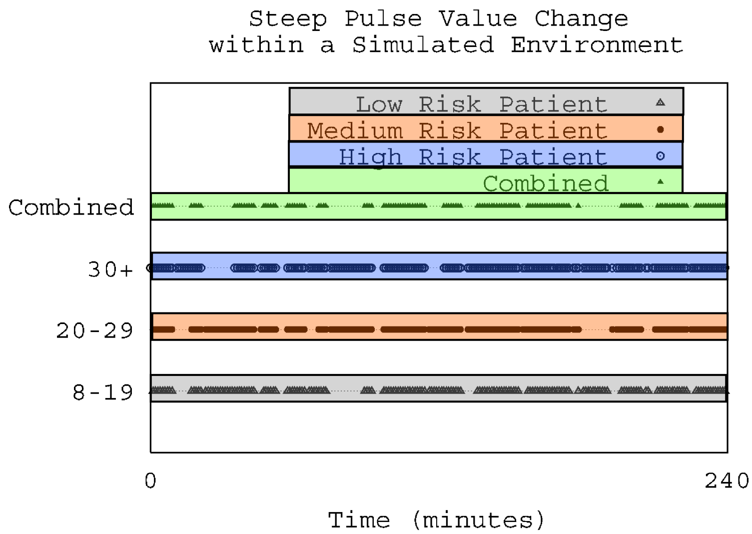


Fig. 6. Steep Pulse Value Change.

Table III. Steep Pulse Value Change. A Patient's Pulse is Monitored over a Three-Minute Period. If it Increases or Decreases by More than or Equal to 8 BPM then Data is Transmitted

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

5.5 Summary

Presented, is the DMS-DCM (Data Management System-Data Consistency Model). It is designed to organize 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 that may reside on a medical practitioner's mobile device is associated with specific members of the medical staff. This ensures that any context update that occurs within the mobile device is transmitted to key members of the 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.

6. CONCLUSIONS

Pervasive environments are complex and dynamic in nature. It is not uncommon for such data-rich domains to be made up of disjointed and unorganized 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 paper 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 that 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 embedded patient sensing device enabled real world experiments to be conducted. It also helped in the design, development, and evaluation of the DMS components.

6.1 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 summarized 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 Section 4, is the Data Management System-Validation Model (DMS-VM). It helps to identify and remove erroneous or invalid data elements within a patient's 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 results comparable to larger, more expensive standard bedside medical devices. It also demonstrates the potential of monitoring patients at home in a nonintrusive noninvasive manner.
— An agent-based data consistency infrastructure is presented in Section 5. The Data Management System-Data Consistency Model (DMS-DCM) utilizes all known information (e.g. user profile, medical practitioner's schedule, and data priority) to contextually synchronize 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.

Pervasive applications require intelligent context management tools. Software agents have been shown to effectively rationalize 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 mobile users.

6.2 Future Work

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

The sensor validation techniques presented in Section 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 any way. 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.

Context-sensitive quality data management architectures can play an important role in utilizing the hardware and software resources within a pervasive computing environment. The DMS presented in this article 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 higher levels of data excellence.

ACKNOWLEDGMENTS

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.

REFERENCES

ANGROVE, P., O'FLYNN, B., BARTON, J., GONZALEZ, A., O'MATHUNA, C., O'DONOGHUE, J., HERBERT, J., AND MIN, S. J. A. 2007. Biomonitor implementation for E-Health. In *Proceedings of the 4th IEEE pHealth Conference.*

BANITASA, K. A., GEORGIADIS, P., TACHAKRA, S., AND CAVOURAS, D. 2004. Using handheld devices for real-time wireless teleconsultation. In *Proceedings of the 26th IEEE Engineering in Medicine and Biology Society Annual Conference*. 3105–3108.

BARDRAM, J. E. AND CHRISTENSEN, H. B. 2007. Middleware for pervasive healthcare. White paper. www.pervasivecomputing.dk/publications/Filewmmc2001.PDF.

BARDRAM, J. E. 2004. Applications of context-aware computing in hospital work—Examples and design principles. In *Proceedings of the 19th Annual ACM Symposium on Applied Computing*. 1574–1579.

BALLAGAS, R., BORCHERS, J., ROHS, M., AND SHERIDAN, J. G. 2006. The smart phone: A ubiquitous input device. *IEEE Pervasive Comput. 5*, 1, 70–77.

BARTON, J., O'FLYNN, B., ANGROVE, P., GONZALEZ, A., O'DONOGHUE, J., AND HERBERT, J. 2005. Wireless sensor networks and pervasive patient monitoring. In *Proceedings of the Information Technology & Telecommunications Annual Conference (IT&T).*

BISDIKIAN, C., BOAMAH, I., CASTRO, P., MISRA, A., RUBAS, J., VILLOUTREIX, N., AND YEH, D. 2002. Intelligent pervasive middleware for context-based and localized telematics services. In *Proceedings of the ACM Mobile Commerce Workshop (MobiCommerce).* 15–24.

BUCHHOLZ, T., KUPPER, A., AND SCHIFFERS, M. 2003. Quality of context: What it is and why we need it. In *Proceedings of the 10th Workshop of the HP OpenView University Association (HPOVUA).* 1–14.

CAO, J., CHAN, A. T. S., AND WU, J. 2001. Achieving replication consistency using cooperating mobile agents. In *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW)*. 453–458.

CHAGOYEN, M. AND PASCUAL-MONTANO, A. 2004. Biological data overload? — Defining today technologies for tomorrow. In *Proceedings of the ECAI Workshop on Data Mining in Functional Genomics and Proteomics: Current Trends and Future Directions*. 8–11.

CHOI, N., VENKATESAN, S., PRAKASH, R., AND PRAKASHH, R. 2004. A QoS-aware MAC layer protocol for wireless LANs. In *Proceedings of the International Conference on Performance, Computing and Communications (MWN)*.

CHRYSANTHIS, P., PRUHS, K., AND LIBERATORE, V. 2003. Middleware support for multicast-based data dissemination: A working reality. White paper, WORDS.

DELLA MEA, V. 2001. Agents acting and moving in healthcare scenario: A new paradigm for telemedical collaboration. *IEEE Trans. Inform. Technol. Biomed. 5*, 1, 10–13.

DING, B., SHI, D., AND WANG, H. 2006. Towards pervasive middleware: A CORBA3-compliant infrastructure. In *Proceedings of the 1st International Symposium on Pervasive Computing and Applications.* 139–144.

DRUGGE, M., HALLBERG, J., PARNES, P., AND SYNNES, K. 2006. Wearable systems in nursing home care: Prototyping experience. *IEEE Pervasive Comput. 5*, 1, 86–91.

FENSLI, R., GUNNARSON, E., AND GUNDERSEN, T. 2005. A wearable ECG recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation. In *Proceedings of the IEEE Symposium on Computer-Based Medical Systems*. 407–412.

GEER, D. 2006a. Nanotechnology: The growing impact of shrinking computers. *IEEE Pervasive Comput. 5*, 1, 7–11.

GEER, D. 2006b. Pervasive medical devices: Less invasive, more productive. *IEEE Pervasive Comput. 5*, 2, 85–87.

GOLMIE, N., CYPHER, D., AND REBALA, O. 2005. Performance analysis of low rate wireless technologies for medical applications. *Comput. Comm. 28*, 10, 1266.

HUANG, Z., SUN, C., PURVIS, M., AND CRANEFIELD, S. 2001. View-based consistency and its implementation. In *Proceedings of the 1st IEEE/ACM International Symposium on Cluster Computing and the GRID.* 74–81.

HUEBSCHER, M. C. AND MCCANN, J. A. 2004. Adaptive middleware for context aware applications in smart homes. In *Proceedings of the 2nd International Workshop on Middleware for Pervasive and Ad-Hoc Computing.* 111–116.

KAVIMANDAN, A., SELIGMANN, D., KLEMM, R., JOHN, A., AND GOKHALE, A. 2006. A client-side architecture for supporting pervasive enterprise communications. In *Proceedings of the IEEE Conference on Pervasive Services (ICPS).*

KEARNEY, L., KIDD, M., MILLER, M., SAGE, J., KHORRAMI, M., MCGEE, J., CASSIDY, K., AND GRAY, P. 2006. Utilising handheld computers to monitor and support patients receiving chemothempy: Results of a UK-based feasibility study. *Support. Care Cancer. 14*, 7, 742–752.

KHOUKHI, L. AND CHERKAOUI, S. 2005. A quality of service approach based on neural networks for mobile ad hoc networks. In *Proceedings of the 2nd International Wireless and Optical Communications Networks Conference*. 295–301.

KISAZUMI, K., NAKANISHI, T., KITASUKUA, T., AND FUKUDA, A. 2003. CAMPUS: A context-aware middleware. In *Proceedings of the 6th Joint Workshop on System Development*.

KRAUSE, M. AND HOCHSTATTER, I. 2005. Challenges in modelling and using quality of context (QoC). In *Proceedings of the 2nd International Conference on Mobility Aware Technologies and Applications (MATA)*. 324–333.

LORINCZ, K., MALAN, D. J., FULFORD-JONES, T. R. F., NAWOJ, A., CLAVEL, A., SHNAYDER, V., MAINLAND, G., WELSH M., AND MOULTON, S. 2004. Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Comput. 3*, 4, 16–23.

LUBRIN, E., LAWRENCE, E., AND NAVARRO, K. F. 2005. Wireless remote healthcare monitoring with motes. In *Proceedings of the International Conference on Mobile Business (ICMB)*. 235–241.

MAZZI, C., GANGULY, P., AND KIDD, M. 2001. Healthcare applications based on software agents. *Studies Health Technol. Informat. 81*, 1, 136–140.

MIHAILA, G., RASCHID, L., AND VIDAL, M.-E. 2000. Using quality of data metadata for source selection and ranking. In *Proceedings of the 3rd International Workshop on the Web and Databases (WebDB)*. 1–12.

NORNAM, D. 1998. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer is So Complex and Information Appliances are the Solution*. The MIT Press, Cambridge, MA.

O'DONOGHUE, J. AND HERBERT, J. 2005. Data management system: A context aware architecture for pervasive patient monitoring. In *Proceedings of the 3rd International Conference on Smart Homes and Health Telematics (ICOST)*. 159–166.

O'DONOGHUE, J., HERBERT, J., AND FENSLI, R. 2006a. Sensor validation within a pervasive medical environment. In *Proceedings of the 5th IEEE Sensors Conference.* 972–975.

O'DONOGHUE, J., HERBERT, J., AND KENNEDY, R. 2006b. Data consistency within a pervasive medical environment. In *Proceedings of the 5th IEEE Sensors Conference*. 968–971.

O'DONOGHUE, J., HERBERT, J., AND STACK, P. 2006c. Remote non-intrusive patient monitoring. In *Proceedings of the 4th International Conference on Smart Homes and Health Telematic (ICOST).*

O'DONOGHUE, J. AND HERBERT, J. 2006d. An intelligent data management reasoning model within a wireless patient sensor network. In *Proceedings of the 1st Artificial Intelligence Techniques for Ambient Intelligence workshop (AITAmI).*

O'DONOGHUE, J. AND HERBERT, J. 2006e. Profile based sensor data acquisition in a ubiquitous medical environment. In *Proceedings of the UbiCare Workshop in Conjunction (with IEEE PerCom)*. 570–574.

O'DONOGHUE, J. AND HERBERT, J. 2006f. A QoS data management system within a pervasive medical environment. In *Proceedings of the Doctor Colloquium (in conjunction with PerCom).*

O'FLYNN, B., BARROSO, A., BELLIS, A., BENSON, J., ROEDIG, U., DELANEY, K., BARTON, J. SREENAN, C., AND O'MATHUNA, C. 2005. The development of a novel miniaturized modular platfonn for wireless sensor networks. In *Proceedings of the IPSN Track on Sensor Plaiform, Tools and Design Methods for Networked Embedded Systems (IPSN/SPOTS)*. 370–375.

O'FLYNN, B., ANGROVE, P., BARTON, J., GONZALEZ, A., O'DONOGHUE, J., AND HERBERT, J. 2006. Wireless biomonitor for ambient assisted living. In *Proceedings of the IEEE International Conference on Signals and Electronic Systems (ICSES)*.

O'FLYNN, B., ANGROVE, P., BARTON, J., GONZALEZ, A., O'MATHUNA, C., O'DONOGHUE, J., MIN, S. J., AND HERBERT, J. 2007. Wireless ECG - Biomonitor for implementation forE-Health. In *Proceedings of the 4th pHealth Workshop*.

O'SULLIVAN, T., O'DONOGHUE, J., HERBERT, J., AND STUDDERT, R. 2006. CAMMD: Contex aware mobile medical devices. *Intern. Universal Comput. Sci.* (Special on Pervasive Health Management: New Challenges for Health Informatics.)

PITOURA, E. AND BHARGAVA, B. K. 1999. Data consistency in intermittently connected distributed systems. *IEEE Trans. Knowl. Data Eng. 11*, 6, 896–915.

RAO, A. S. AND GEORGEFF, M. P. 1991. Modeling rational agents within a BDI-Architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*.

ROUSSOS, G., MARSH, A. J., AND MAGLAVERA, S. 2005. Enabling pervasive computing with smart phones. *IEEE Pervasive Comput. 4*, 2, 20–27.

SAHA, D. AND MUKHERJEE, A. 2003. Pervasive computing: A paradigm for the 21st century. *Comput. 36*, 3, 25–31.

SHEIKH, K., WEGDAM, M., AND VAN SINDEREN, M. 2007. Middleware support for quality of context in pervasive context-aware systems. In *Proceedings of PerWare Workshop* (Co-Located with *IEEE PerCom Conference*).

SHIRAZI, B., KUMAR, M., AND SUNG, B. Y. 2004. QoS middleware support for pervasive computing applications. In *Proceedings of the 37th Hawaii International Conference on System Sciences.*

SOLDTOS, J., PANDIS, I., STAMATIS, K., POLYMENAKOS, L., AND CROWLEY, J. L. 2007. Agent based middleware infrastructure for autonomous contextaware ubiquitous computing services. *Comput. Comm. 30*, 3, 577–591.

STAUSBERG, J., KOCH, D., INGENERF, J., AND BETZLER, M. 2003. Comparing paper-based with electronic patient records: Lessons learned during a study on diagnosis and procedure codes. *J. Amer. Med. Informatics Assoc. 10*, 5, 470–477.

SUN, X. H. AND BLATECKY, A. R. 2004. Middleware: The key to next generation computing. *Preface J. Parallel Distrib. Comput. 64*, 689–691.

VARSHNEY, U. 2006. Managing wireless health monitoring for patients with disabilities. *IT Profess. Technol. Solutions Enterprise 8*, 6, 12–16.

WEINSTEIN, E., HO, P., HEISELE, B., POGGIO, T., STEELE, K., AND AGARWAL, A. 2002. Handheld face identification technology in a pervasive computing environment. In *Proceedings of the International Conference on Pervasive Computing.*

WEISER, M. 1991. The computer for the 21st century. *Sci. Amer. 265*, 3, 94–104.

WINTERS, J. M. AND WANG, Y. 2003. Wearable sensors and telerehabilitation. *IEEE Eng. Med. Biol. Mag. 22*, 3, 56–65.

WOOLRIDGE, M. 1997. Agent-based software engineering. *IEEE J. Softw. Eng. 144,* 1, 26–37.

ZASLAVSKY, A. 2004. Mobile agents: Can they assist with context awareness? In *Proceedings of the IEEE International Conference on Mobile Data Management (MDM)*.

Received February 2012; accepted June 2012