An Adaptive Middleware Applied to the Ad-hoc Nature of Cardiac Health Care

Gemma Power, Christopher Foley, Sasitharan Balasubramaniam, Dimtri Botvich
Telecommunications Software & Systems Group
Waterford Institute of Technology
Waterford, Ireland
+35351302981
{gpower, ccfoley, sasib, dbotvich}@tssg.org

ABSTRACT
Heart disease is the number one killer in the civilized world, accounting for around 1.9 million people every year in the EU, with the associated annual health costs of approximately EUR 105 billion. Important aspects of treating cardiac problems is monitoring and directing critical data to key individuals. The middleware proposed in this paper interacts with ECG sensors and provides a dynamic decision making framework for forming critical decisions when defined thresholds are exceeded. The middleware and its defined services target the embedded device domain which has an existing large scale deployment within the healthcare sector. The services help to ease development and address problems like dynamic grouping, load balancing and providing a uniform level of abstraction from the underlying network.

Categories and Subject Descriptors
D.2.11 [Software Architectures]: *Domain-specific architectures.*

General Terms
Algorithms, Management, Measurement, Experimentation, Human Factors.

Keywords
Middleware, groups, services, policies, health care management.

1. INTRODUCTION
Staff in the healthcare sector are expected to provide quality care to an increasing number of patients while using a limited amount of resources. To realize a proficient service, healthcare must utilize technology. Mobile and ubiquitous computing is becoming more prevalent in everyday life; harnessing this and its endless capabilities will be an important component in providing a fitter more pervasive healthcare system.

The development of a middleware, which is essentially the glue between the mobile embedded devices, which hides the complexity of the underlying network and eases software development for software engineers, is central to solving the puzzle.

The solution proposed in this paper is a middleware which attempts to address the challenges discussed previously. It targets mobile embedded devices, exploits the vast potential of Body Area Networks (BANs) [8]and is realized and demonstrated in the health care domain. A BAN is based on technology for monitoring and logging of vital signs and supervising the health status of patients who suffer from chronic conditions like diabetes or cardiac problems. They consist of a compact hardware unit physically attached to the patient, which is capable of both sensing the data and transmitting it to the necessary medical personnel for processing.

The paper is structured as follow; Section 2 introduces related work. Section 3 discusses the scenario. Section 4 introduces the middleware architecture and section 5 is a complete case study.

2. RELATED WORK
The recently established "Heart Cycle consortium" was setup by several EU partners to improve the quality of life for cardiac patients by monitoring their condition [4].

A number of other approaches have examined problems in the healthcare scenario, the idea of improving healthcare through technology and the related research challenges are examined in [13]. The adaptation of policies for the healthcare environment has been examined by the Amuse project [6] . Controlling services by using health care specific policies has been investigated by [7]. The reorganization of decision making groups by dynamically changing who makes the decisions for a particular goal are studied by [2]. Our proposed solution differs from the above methods as policies are used to control the dynamic reconfiguration of groups, allowing for a more intelligent use of medical resources and staff. It builds on top of technologies already available like sensors for example and provides processed data for the end user. Instead of an off the shelf Middleware solution we propose a middleware solution that can be easily adapted to suit any environment. This approach provides patients with more efficient and effective care.

3. SCENARIO
Heart disease is a major killer [14] and it uses up vast medical resources in its treatment. Symptoms of heart failure include chest pains and shortness of breath. There are numerous suspected causes of heart disease, diet and family history being the two obvious culprits, another contributor is high blood pressure or Hypertension as it's known in the medical world. As many

medical environments experience staff and resource shortages a more efficient management system for patient monitoring is required. This would provide the medical establishment with effective load balancing of staff by using technology to identify a medical professional's immediate workload, area of expertise as well as their location and managing them accordingly, thus reducing the negative impact of resource shortages. As the cardiology area is classed as critical, real time information reporting on a patient is crucial, here technology can also provide a solution in the form of a Body Area Network (BAN). A BAN is a key component in the supervision of patients suffering from grave diseases like diabetes or cardiac problems. They are made up of hardware sensing/transmitting modules which are physically attached to the patient's body.

Problems in the healthcare sector can often be complex issues and are rarely solved in full by a single individual, normally a group of medical professionals working together are required to resolve the issue. This had led to the emergence of the 'group' concept as a critical utility, where a patient or set of patients and an assortment of medical staff can be "grouped" together for both monitoring and treatment purposes to enhance the care.

A contributor to achieving this more efficient care is the application of the MORE middleware [1]. By employing technology different monitoring groups can be set up and modified on an ad-hoc basis. These groups could consist of; a patient, nurse and doctor or just a patient and nurse if the sensor readings don't initiate an alarm indicating abnormal data and the patient needs to be examined by a doctor. This scenario allows for the effective remote monitoring of a cardiac patients' critical data in addition to effectively balancing the workload of the available doctors. This could be a potential solution to the negative impact of resource deficiencies in medical environments.

4. ARCHITECTURE

4.1 Overall System Architecture

In the overall system architecture service interact and extract data from low level sensors (e.g. Body Area Networks) and other data sources as shown in Figure 1.

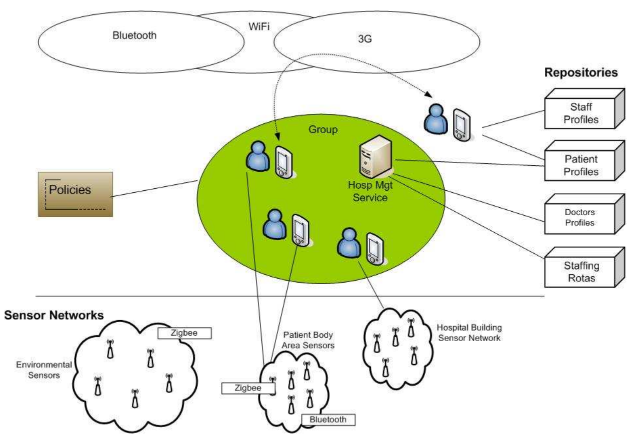

Figure 1. Overall System Architecture

These services can be grouped based on their functionality to form a communication group which can achieve some defined task. Services interact through whatever communication medium is available to them, e.g. WIFI, Bluetooth, 3G. The main advantage of this setup is that groups can be created on an ad-hoc basis providing a more effective care structure for patients as well as maximizing the resources that are available. The MORE middleware will be located on the respective user devices for example, PDA or Laptop. An example of this architecture is portrayed in Figure 1.

4.2 Middleware Architecture

The system architecture is largely based on a middleware architecture detailed in the European funded project – Network centric Middleware for group communication and resource sharing across heterogeneous embedded systems (MORE) [1], a diagram depicting the architecture is shown in Figure 2.

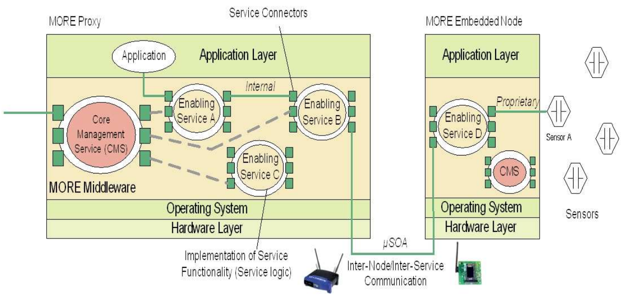

Figure 2. MORE Middleware Architecture

As can be seen from 2 the MORE architecture consists of four core layers; the application layer, the MORE Middleware, the operating system and the hardware layer. The architecture to a certain degree has a traditional layered approach but the central middleware section is based on the Service Orientated Architecture (SOA) paradigm. The central component to a MORE Middleware deployment is the Core Management Service (CMS) which is a supervision service for the other deployed services. The MORE architecture is based on a number of different services deployable on devices as diverse as embedded and servers. The CMS is the key part of the MORE middleware because it provides the generic functionality required for the administration of the Middleware. All MORE services are composed of service connectors which provide communications between services, the implementation of the service interface and, the actual service logic itself.

Connectors provide communication between remote as well as local services using a range of protocols like SOAP, Direct Java (inter thread communication), TCP sockets and, uSOA. The uSOA [10] connector is particularly specific to the requirements of MORE for embedded devices as it has a reduced message size and thus cuts down on XML parsing. With the SOA paradigm, connectors are separated from components, keeping computations and interactions separate throughout a system [9]. The MORE architecture also allows for proprietary connections between services and specific sensors. The architecture is extendable in that a new protocol can be included as a connector.

4.2.1 *Adopted Technologies*

MORE services are built on the concept of DPWS (Device Profile for Web Services) [5] . DPWS defines a set of specifications to allow secure Web Service messaging, discovery, description, and eventing on resource-limited devices. DPWS is fully compliant with Web Services. SOAP as previously mentioned in the connector context is the standard used in Web services because of attributes such as adaptability and extensibility provided by XML. The service interface was designed using a Web Services Discovery Language (WSDL) file [11]. This WSDL defines the, elements, types operations and bindings for the service. The WSDL file is fed into a DPWS generator which creates a set of Java files including a Notifier and Invoker critical for sending and receiving messages which eases the life of the software developer.

4.2.2 *Group Services*

Group Services are a collection of services for improving the middleware's usability for developers as well as users. The purpose of group services is to provide the ability to statically and dynamically create different groups in addition to providing a system which administers and manages these groups.

Groups are controlled by policies, policies are used for configuring and building groups, in addition to dynamically adding new members to a group.

The policy has the following structure: *{Event, Condition, Action} [15]*. A more detailed explanation on how the policies are used to control the groups is described in [3].

5. CASE STUDY

In an attempt to prove the benefits of technology in the Healthcare domain a case study was undertaken. This case study is a realization of the Body Area Networks theory where a mobile compact wearable sensor is attached to a human body to remotely monitor vital body data like ECG readings. The main components are a circuit which was built to record ECG data, and this is backboned by a number of middleware services representing the different roles of various actors in cardiac healthcare.

The key point which the case study portrays is the rearrangement of groups of middleware services driven by real time medical data.

5.1 ECG Circuit

An electrocardiogram is a graph produced by an electrocardiograph, which records the hearts electrical activity via electrodes connected to the body's surface. An ECG provides vital information like a person's heart rate. The heart rate can be easily recognized on the graph as it is determined by the distance between "QRS" peaks (Figure 3) for a healthy adult it is normally between ~ 70 – 130 beats per minute. This information is crucial to effective monitoring of a patient with a heart condition. The ECG reading can be used to detect numerous cardiac problems including Hypertension. As ECG's are not readily available for purchase it was necessary to build one. This was achieved by assembling a circuit which contained a number of Operational Amplifiers to take in the signals from the electrodes and provide an output for analogue to digital conversion.

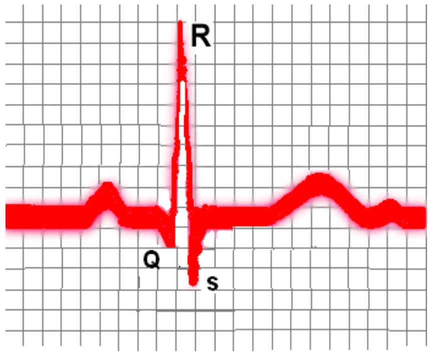

Figure 3. QRS Wave

The circuit also contained several low power resistors, capacitors and diodes for safety. The IN- and IN+ were connected to our patient's chest (Figure 4) and the BODY electrode was connected to the patients arm, this provided the necessary feedback for the circuit. This circuit is powered by a 9V supply and the output from IC3B is fed into the analogue to digital converter of our "Squidbee Sensor" [2].

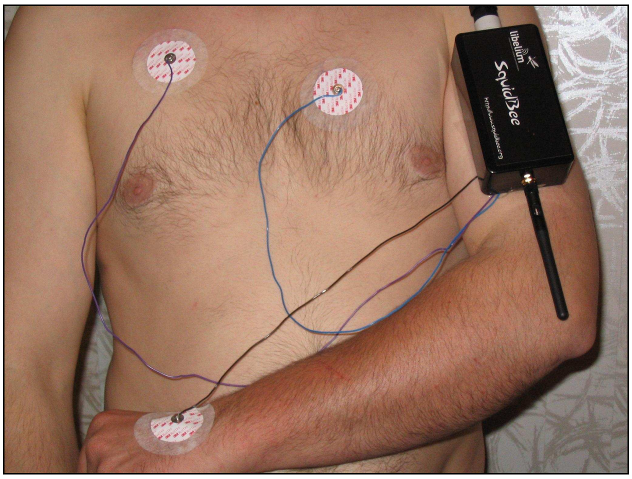

Figure 4. ECG Connected to Patient

Figure 4 shows the ECG circuit attached to a patients body and the Squidbee device which is used for wireless transmission of the recorded data.

5.2 ECG Middleware Service

The Squidbee sensor is programmed to transmit the ECG readings in real-time to any listening Zigbee [16] enabled device. Zigbee is the specification for a set of communication protocols that use low-power digital radio transmissions. It is based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs), for example wireless headphones set connecting with a mobile phone via short-range radio. The technology is intended to be simpler and cheaper than Bluetooth. Zigbee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking, the operating frequency of Zigbee is 2.4GHz. A Squidbee gateway is required to transfer the data that is being sent via the sensors to a computer or laptop, this is connected to the computer's USB or serial port. An illustration of the setup is shown in Figure 5.

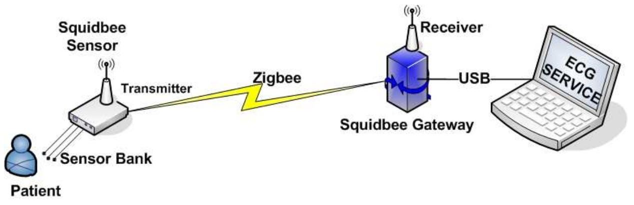

Figure 5. Patient Connection Setup

A MORE middleware service was developed and deployed on a laptop [1], which read the ECG data from the serial port. In order to implement this connection using Java, a driver called "RxTxComm" [12] was required.

5.3 Initial Setup

In our case the parties involved are; two patients, (Patient A and Patient B) a nurse and a doctor. The patient is connected as specified in Figure 5. The ECG middleware service is running on a laptop. The Nurse has a Medical Graphing Service running on her device and the doctors have a Physician service running on their devices. The test setup includes a device with GMS running on it which enables group communication. The initial setup is displayed in Figure 6. Two groups exist, one for each patient.

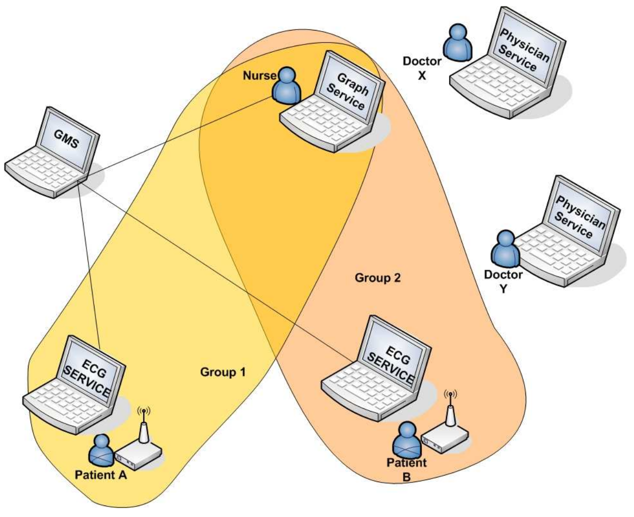

Figure 6. Initial Group Configuration

Each patient has a heart condition and therefore needs to be closely monitored by medical staff. There is periodic transmission of data from the sensor to the group members. The GMS forwards these readings to the members of the patients group.

Both patients can be remotely monitored by the nurse's service which graphs the incoming ECG readings. Figure 7 shows readings from Patient A which has a normal heartbeat.

The sampling rate of the sensor was programmed to be 0.2 seconds, this means the sensor transmitted the sampled ECG readings every 0.2 seconds. To calculate the Heart rate in beats per minute we used the following equation;

> Heart Rate (HR) = Sampling Rate intervals (SI) / 300 (60/0.2) where I = intervals between R's (peaks shown on the graph) previously described in Figure 3.

For Patient A this equation gave a HR of; 300/4 = 75 beats per minute.

Figure 7. Normal ECG Readings (Patient A)

In our case study the Nurses service is developed to graphically display the ECG readings as well as an alarm that detects whether the patient is at risk of Hypertension (Hypertension is the medical term for high blood pressure). Hypertension readings are measured in "millimeters of mercury (mmHg), they are normally given as two numbers for example 150 over 95; the first number is your systolic pressure (the pressure created when your heart beats). The second number is the diastolic pressure (the pressure inside the blood vessels when the heart is at rest). Hypertension is considered to be present when a person's systolic blood pressure is consistently 140 mmHg or greater, and/or their diastolic blood pressure is consistently 90 mmHg or greater. Hypertension can be detected by consistently high amplitude readings on an electrocardiogram (ECG).

5.3.1 Policy Rule Distribution

When a group is created the group members (i.e. services) can only receive a predefined set of messages or events, which is defined in the service's WSDL file. This is in turn a precursor of what events the group member will be forwarded during group communication. Each policy rule is triggered by an event (or message), therefore the policy rules which are relevant for each group member are distributed to them at group creation. The Policy Extract 1 below shows two rules which are distributed to the MedicalGrapherService on the Nurse's device. The MedicalGrapherService was programmed to detect high "R" amplitude readings.

```
PolicyRule PR1: {
PolicyEvent:E1
{CommonName "NewSensorData"
 Parameter "ECGValue" }
PolicyCondition1:{CommonName "ECGHighPriority"
 If ECGValue > 400 }
AND
PolicyCondition2: {CommonName "HighQRSflagSet"
 If HighQRSflag == true }
PolicyAction2: {CommonName " HypertensiveAlarm",
TriggerMsg "SendAlarmToGroup"} }

PolicyRule PR2: {
PolicyEvent:E1 {CommonName "NewSensorData"
 Parameter "ECGValue" }
PolicyCondition1: {CommonName "ECGHighPriority"
 If ECGValue > 400 }
```

```
PolicyAction1: {CommonName ”SetHighQRSFlag”,
 set HighQRSFlag = true} }
```

Policy Extract 1 Policy Rules for Hypertension Detection

Policy Extract 1 shows two policy rules which are triggered by the *NewSensorData* event. Hypertension is detected by two consecutive high QRS values, as discussed in Section 5.3. Policy rule *PR2* sets the initial high value flag. The second policy rule PR2 detects consecutive high readings and triggers an Hypertension alarm to be sent to the group members.

5.4 Reconfiguration based on Efficiency of Services

The alarm generated for Hypertension detection is sent to the group members. GMS will receive this message and check to see what policy rules should be evaluated. Policy Extract 2 is a sample of the rule which is evaluated within GMS.

```
PolicyRule PR3
PolicyEvent:
{CommonName ”HypertnsionAlarm”,
ParameterList ”PatientIdentifier”, ”ECG-Value”}
PolicyCondition: {CommonName ”hasGrpMemberDoctor”,
If !(GrouphHasMember == Doctor) }
PolicyAction: {CommonName ”addMemberToGroup”,
TriggerFunctionality ”addMemberToGroup”
Parameters ”{MemberType, Docotor}”,
{SelectionBasedOn, efficiency}} }
```

Policy Extract 2 Policy rule for Group Reconfiguration

Policy rule PR3 is triggered by the HypertensionAlarm, it checks to see if the group has a member of type doctor and if not it triggers the addMemberToGroup functionality.

5.4.1 Service Efficiency Criteria

GMS uses an algorithm to determine the “fittest” doctor to add to the group. Firstly a number between 0 and 1 is afforded to each doctor with a weighting on cardiology expertise. So for example a diabetologist might have an expertise value (E) of 0.2 (remembering that the alarm is triggered by a heart condition) and a cardiologist might have an E of 0.9. The algorithm also equates each doctor’s current workload (WL) to ensure that the doctor chosen is available to deal with the alarm. The fitness doctor “F” is then calculated from:

$$F = \alpha E + 1/WL$$

When the HypertensionAlarm is received by GMS the above algorithm is used to make the best selection as to who to assign the patient to. In our example, Doctor X is chosen. GMS will add Doctor X’s PhysicianService to the Group 2 and all messages sent to the group is now sent to both the Doctor and the Nurse. This can be seen in Figure 8.

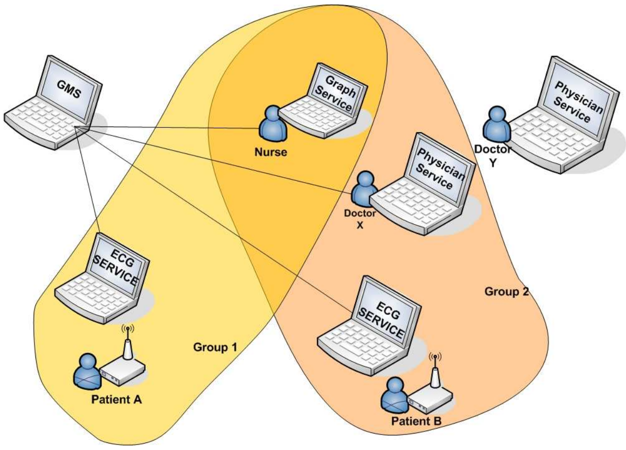

Figure 8. Reconfigured Groups (Group 2 includes Doctor)

Once the doctor has been added to the patient and nurse group they automatically receive the transmitted ECG readings. Figure 9 is sequence diagram showing the flow of information for Group 2

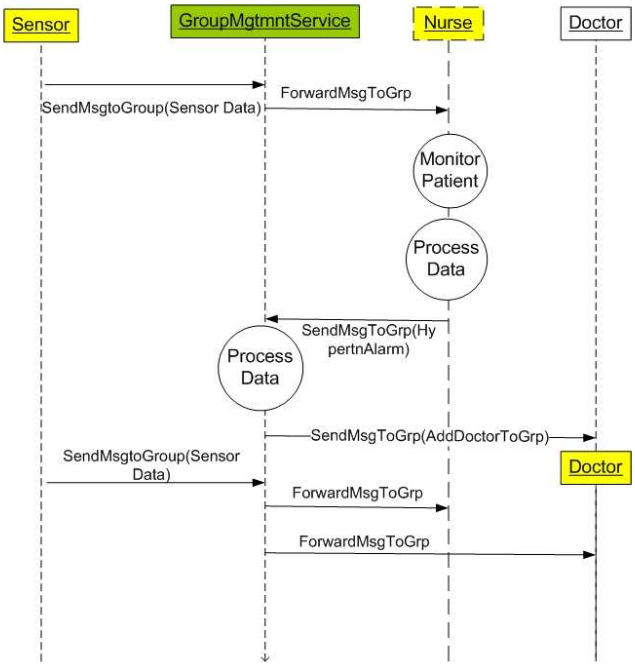

Figure 9. Group Reconfiguration Flow

As can be seen from the yellow box on the right of this diagram the doctor becomes part of Group 2 after an alarm has been triggered, from this point on both the doctor and nurse will receive any SendMsgToGroup messages from GMS.

Figure 10 shows an example of an ECG reading from a patient with hypertension.

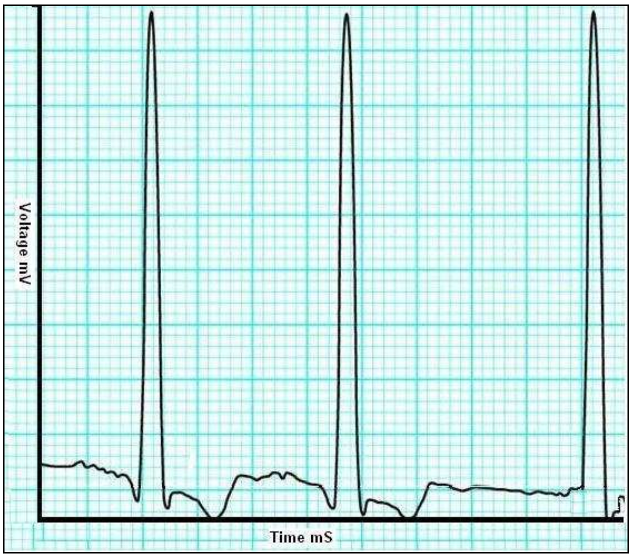

Figure 10 Abnormal ECG Reading (Hypertension)

6. CONCLUSION

The fight against heart disease and race for pervasive technological solutions which ease patient monitoring presents a tremendous challenge. This paper looks at a middleware deployment in the form of services which can build on the advancement in sensors in addition to assembling individual services into groups. The middleware is the basis of an adaptive framework which hides underlying network complexity and simplifies the development process. The use and application of the middleware can provide a more effective monitoring management system for the healthcare environment, allowing for patients to be remotely observed, perhaps from the comfort of their own homes. Consequently this alleviates the huge pressures placed on hospitals to make beds available. Furthermore doctors workloads are also taken into consideration and doctors are allocated to groups accordingly.

7. ACKNOWLEDGMENTS

The work described in this paper was partly funded by the European Union through the MORE project in the 6th Framework Programme under the Reference-ID FP6-IST-032939.

8. REFERENCES

[1] A. Wolff, S Michaelis (Editors): "MORE System and Architecture" , Public Deliverable, EU-Project MORE, Dortmund 2007, http://www.ist-more.org

[2] Barber et al. 2001. Dynamic reorganization of decision-making groups. International Conference on Autonomous Agents.

[3] C Foley et al 2008, Distributed Pervasive Services using Group Service communication supporting Body Area Networks.

[4] Compliance and effectiveness in HF and CHD closed-loop management" HeartCycle project. http://www.heartcycle.eu/

[5] Device Profile for web services specification, http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf

[6] E. Lupu , N. Dulay , M. Sloman , J. Sventek , S. Heeps, S Strowes, K. Twidle , S.-L. Keoh , A. Schaeffer-Filho, "AMUSE: autonomic management of ubiquitous e-Health systems", Proceedings of the UK e-Science All Hands Meeting 2005, Nottingham UK.

[7] Feng Wang, Liam S. Docherty, Kenneth J. Turner, Mario Kolberg, Evan H. Magill. Services and Policies for Care at Home. In Jakob E. Bardram, Juan Carlos Chachques and Upkar Varshney, editors, Proc. 1st. International Conference on Pervasive Computing Technologies for Healthcare, pages 7.1–7.10, Institution of Electrical and Electronic Engineers Press, New York, USA, November 2006.

[8] German Ministry for education, 2002. Body Area Network (BANs) http://www.ban.fraunhofer.de/index_e.html.

[9] N. R. Mehta et al, 2000. "Towards a Taxonomy of Software Connectors".http://sunset.usc.edu/classes/cs599_2000/Conn-ICSE2000.pdf

[10] Newcomer, Eric; Lomow, Greg (2005). *Understanding SOA with Web Services*. Addison Wesley. ISBN 0-321-18086-0.

[11] Olaf Zimmermann et al, 2005. Perspectives on Web Services: Applying SOAP, WSDL and UDDI to Real-World Projects.

[12] RXTX Website "http://www.rxtx.org/"

[13] Upkar Varshney, "Pervasive healthcare and wireless health monitoring" Mobile Networks and Applications, Volume 12, Issue 2-3 (March 2007)

[14] WHO figures for heart disease deaths 2007. http://www.who.int/cardiovascular_diseases/en/cvd_atlas_14_deathHD.pdf

[15] Wood, P. et al "Event Condition-Action Rule LanguagesFor the Semantic web"

[16] Zigbee Alliance "http://www.zigbee.org/en/index.asp"