Distributed Pervasive Services using Group Service communication supporting Body Area Networks

Christopher Foley[1], Sasitharan Balasubramaniam[1], Dmitri Botvich[1], William Donnelly[1], Stefan Michaelis[2], Jens Schmutzler[2], Thomas Stair[3]

[1]Telecommunications Software & Systems Group
Waterford Institute of Technology
Waterford, Ireland
+353 51 306143
{ccfoley,sasib,dbotvich,wdonnelly}@tssg.org

[2]Communication Networks Institute
University of Dortmund
Dortmund Germany
+49 213 755 3781
{stefan.michaelis,jens.schmutlzer}@udo.edu

[3]Brigham and Women's Hospital, Department of Emergency Medicine
75 Francis Street,
Boston MA
02115
tstair@partners.org

ABSTRACT

Body Area Network (BAN) provide critical data in healthcare monitoring environments, where such monitoring can be performed in a ubiquitous manner using various miniature device technologies. However, a key requirement in supporting the full capacity of a BAN is an efficient distribution, processing and application of the acquired data. The architecture and applications which capitalize on the huge potential of this data, provide significant added value to BANs. This paper proposes an architecture which is service oriented and integrates the data produced by BANs into a healthcare environment, supporting remote interactions between medical officers to maximise patient care. The dynamic interaction of distributed services in this diverse environment is a key ingredient in the way technology can enhance healthcare. The architecture defines group services which facilitate the control of the dynamic behaviour of services within this heterogeneous environment.

Keywords

Policies, groups, services, health-care, monitoring

1. INTRODUCTION

Healthcare workers are under increasing pressure to provide more efficient services to more people using limited resources in today's environments. In order to satisfy these demands, healthcare needs to look towards technology for a potential solution, specifically in an attempt to achieve pervasive healthcare. Continuing technological advances in engineering and communication technologies have paved the way for a new generation of embedded wireless devices and in particular BANs. The BAN is a key system that monitors a patient's condition and provides vital data that can be processed by medical officers, therefore it is a key facilitator in the attempt to achieve pervasive healthcare. Although, while substantial attention has been focused on sensor developments, very limited emphasis has been placed on the processing of the information that will enable efficient healthcare systems. This is especially crucial for hospitals with large number of patients and limited hospital staff.

A key facilitator in achieving pervasive healthcare is through establishment and interaction of medical based services. The grouping of these services and the way in which the groups interact is pivotal to realizing an added value. Based on this group service behaviour, collaboration and organization of multiple people, devices and data sources will enhance the capability of healthcare.

The solution proposed in this paper addresses mechanisms that allow services to be dynamically grouped and to interact with each other based on information gathered from the BAN. The grouping of services will take into consideration, the condition of the patients, types of medical attention required, as well as expertise of the medical officers. Services and their groups can reorganize themselves dependent on events occurring in the system (e.g. emergency – sudden change in a particular's patient electrocardiogram (ECG)). The reorganization and dynamic behaviour is determined by a *Policy Based Management (PBM)* system. The solution comprises of two sets of policies, which include: (i) configuration policies that govern the initialization and configuration management of the groups, and (ii) behaviour policies which are derived from the disease profiles that provide medical logic required to evaluate the condition of patients as well as support interaction between medical officers. The proposed solution can be applied in a home or hospital environment. The case study in this paper highlights its use in the hospital environment, where the focus is on setting up the groups as well as reorganization of the groups in the event of an emergency.

BodyNets'08, March 13–15, 2008, Tempe, Arizona, USA.
Copyright 2008 ISBN 978-963-9799-17-2

This paper is organized as follows: Section 2 introduces related work. Section 3 presents the architecture of the middleware used for evaluating data from a BAN as well as the grouping of services, while section 4 presents the case study. Lastly, section 5 concludes the work described in this paper.

2. Related Work

The vision of pervasive healthcare through wireless technology and the associating research challenges are discussed in [6]. A number of the research challenges are addressed in this paper, namely autonomous and adaptive behaviour of the system. The adaptation of policies as a tool to implementing pervasive healthcare has been used by the Amuse project [3]. Amuse have assigned policies to each device and the role which the device plays within their Self-Managed Cell. Control of services using policies in pervasive health care in the home environment has been investigated by [4]. Our proposed solution differs from these approaches in that policies are used to control services but within the context of a specific group, therefore a service can be a member of different groups and can have different behaviour based on different policies. The proposed solution also associates the policy with a disease variant and a set of policies, both for configuration and behaviour, are derived from this.

The W3C WS-PolicyFramework [9] contains a number of specifications which describe a mechanism for defining and applying policies to a service. The WS-Policy of web service specifications allows services to describe functional assurances which services expect from and provide for callers. These functional assurances are in the context of security, transactions and reliable messaging. These policies are applied to each service and could be applied to a MORE service also but the polices discussed in this solution (Section 3.1.1) are applied to a logical group of services, which provide semantics at a group level.

Context aware middleware services as described in [7] provide a potential data source which could be used to influence group formation and manipulation within the policy structure of the proposed solution.

3. Architecture

The overall system architecture is built upon a middleware architecture specified in the European funded project – Network-centric Middleware for Group communication and Resource sharing across heterogeneous embedded systems (MORE) [2]. The MORE middleware architecture, illustrated in Figure 1, follows the approach of Service Oriented Architectures (SOA).

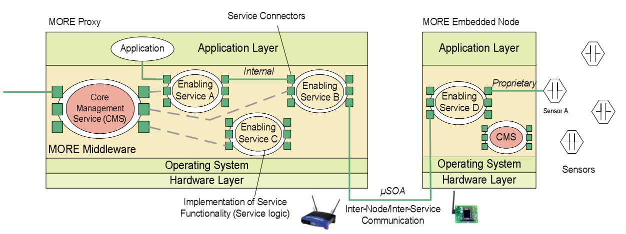

Figure 1. MORE Middleware Architecture

The MORE middleware will provide a set of services offering basic functionality as well as abstract services, which can be used as templates for easier implementation of application specific functionality (e.g. creation of medical logic or addition of new logic to existing services supporting a particular disease or condition). Besides providing a set of services, the MORE architecture also supports an interface to external sensors for gathering information. All services provided by MORE are classified based on the functionality they provide, e.g. Communication Services, Group Services etc.

The overall system architecture is illustrated in Figure 2, where the services are organized into groups, gathering information from BANs. In the system users connected to various wireless networks (e.g. 3G, WiFi), are able to connect virtually through group services, which in turn are collecting data from sensors and feeding it to the groups. The group services in turn will interconnect various medical officers who will evaluate the information. For example, a group can be formed between a doctor, nurse, and a patient with BANs. The nurse will be in charge of monitoring the data closely, while periodic updates will be transmitted to the doctor to give a snapshot of the patient's current condition. Such applications, can allow group services to dynamically process information from BANs for patients in hospital waiting rooms (e.g. SMART project [5]), or patients residing at home, or in transit in an ambulance. At the same time, applications of this architecture can support emergency relief for large scale catastrophes. The MORE middleware will reside on the different user devices and servers and possibly on sensors, dependent on their capabilities.

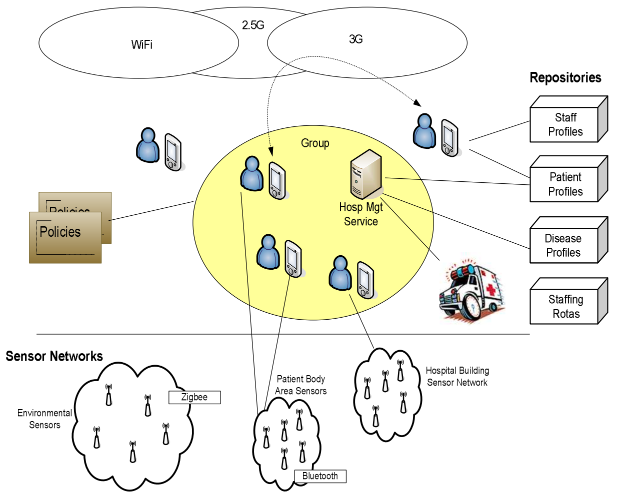

Figure 2. Overall System gathering information from BAN

3.1 Group Services

Group Services are a set of services which will aid and enhance the usability of the middleware for both the developer and user. They provide the means to statically and dynamically build groups and also provide a mechanism through which the behaviour and management of these groups can be controlled. Moreover, they provide the means to send a message to a group in the most efficient manner.

There are specific group types which must be introduced to aid the understanding of effective group establishment. These group types are:

- Role Group - A grouping which consists of actors which have the same primary role in the system, e.g. Nurses
- Service Group - A grouping of services which provide the same functionality in the system.
- Communication Group - A grouping of possibly non related members which need to be brought together in order to complete a specific task, e.g. Specific Patient Monitoring Group

Members of both role and service groups provide the main building blocks for a communication group. Dependent on what goal or task a communication group is trying to achieve, it may result in different combinations of services and individuals being brought together.

The architecture provides two Group Services which act as utility services which expose group functionality to other services:

- Group Management Service (GMS) – management functionality, policy handling and evaluation, storage of group details
- Group Communication Service (GCS) – mechanism for efficient message distribution to group members

Group Services provide a point of convergence through which large numbers of devices can be conveniently manipulated.

3.1.1 Policy Controlled Groups

Each group is controlled by their own configuration and behaviour policy. The use of policies within the Group Services is based on the use of policies in the Policy Based Management domain. Policies are used to handle groups under two different aims, (i) as an aid when configuring and building groups, and (ii) as a way to control the behaviour of a particular group.

The policy has the following structure:

{Event, Condition, Action}

Where the elements meanings are:

- *Event* - An occurrence of an important message/incident which can be used to trigger the evaluation of Conditions
- *Condition* - An aggregation of individual conditions which define the prerequisites for resulting actions to be taken or not.
- *Action* - Represents the necessary actions which need to be taken if the Condition evaluates to true.

All Events, Conditions and Actions are defined as standalone entities in the policy file. When specific instances of the three are combined into one overall entity *{Event, Condition, Action}*, it is known as a PolicyRule. It is the PolicyRules which dictate the behaviour of the group. Figure 3 illustrates the relationship between policies and group instances and where they reside in the system. The policy files themselves are stored in XML format in the GMS service. A group instance is created by issuing a request to GMS, therefore the details of that group instance is stored in GMS for the lifetime of the group. Part of group creation is the mandatory assignment of a configuration and behaviour policy to the new group. This mapping also resides in GMS. The configuration policy impacts the group's setup as a whole. The behaviour policy impacts the run time actions which the individual members must take within the context of the group.

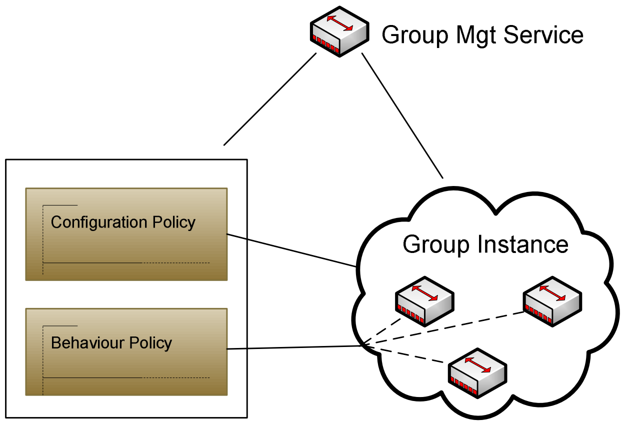

Figure 3. Policy Group Relationship

3.1.2 Group Service Interaction

The Group Services, both GMS and GCS are utility services which can be used to implement group functionality. Figure 4 illustrates the interaction between services within a group.

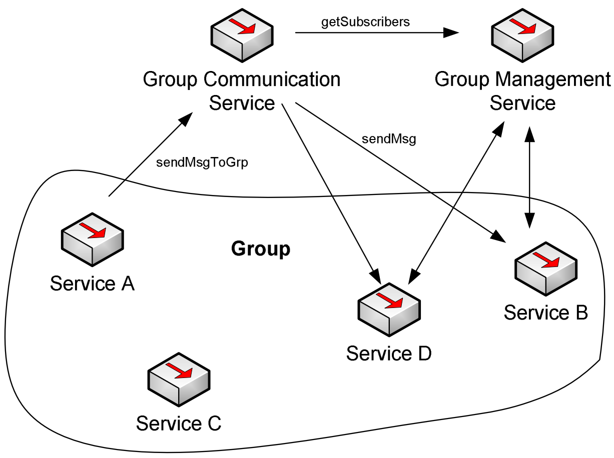

Figure 4. Service Interaction with Group Services

Consider a group with a number of services. Service A wishes to send a message to the group. It will send the message directly to GCS. GCS will query GMS to access the list of subscribed group members for this message type. On receiving this GCS will send the message to all subscribed members (Service C is unsubscribed to the message type).

Once the message is determined as group related by a receiving service, e.g. Service B, it needs to check the policy for that group to determine what actions to take. The policy needs to be parsed and put into a machine readable format. This can be done at two different times depending on the capabilities of the device which the service resides on:

1. at group formation – the service is informed by GMS that it is now part of a group, the service can request a parsed version of the associating group policy which it in turn caches locally.
2. on receiving a group message – the service can at this point request a parsed version of the associating

PolicyRules (i.e. not the complete policy) which match the incoming message type from GMS.

Service B and D will then evaluate the conditions in the relevant PolicyRule and trigger the action if the evaluation has a positive result.

The flow of messages between services within group communication for the second approach is depicted in Figure 5.

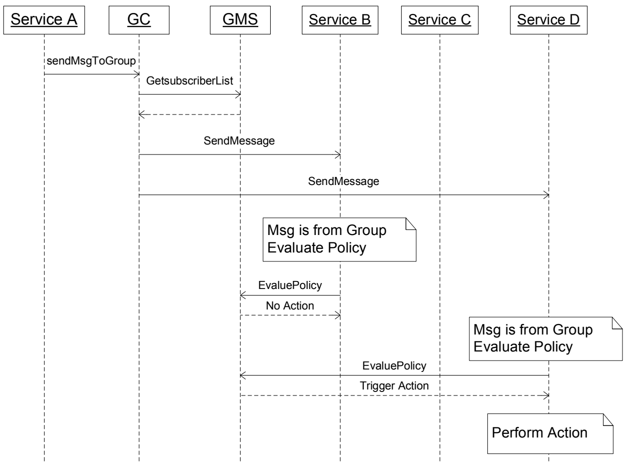

Figure 5. Message Passing in Group Communication

3.1.3 Group Services and Security

Due to the primary target domain being healthcare and the transmission of personal and sensitive data, security becomes a key system requirement. The MORE platform [2], provides a set of security services which aid the other services in meeting the security requirements. These services include: Cryptographic Service, Identity provider Service, Permission Management Service and a Key Distribution Service, all detailed in [1]. The group services specified in the previous section use these security services to provide functionality like: message encryption, message authentication, non-repudiation, user authentication and permission management.

3.1.4 Disease Model

Disease management will involve a detailed database of diseases which is modeled in a way that provides services the means to aid medics when treating and evaluating the diseases. The disease model is based on identification of high level disease and breaking them down into sub-variants of those diseases as well as relationships between the different variants. Figure 6 illustrates an example of an extract of the model with different variants for Diabetes.

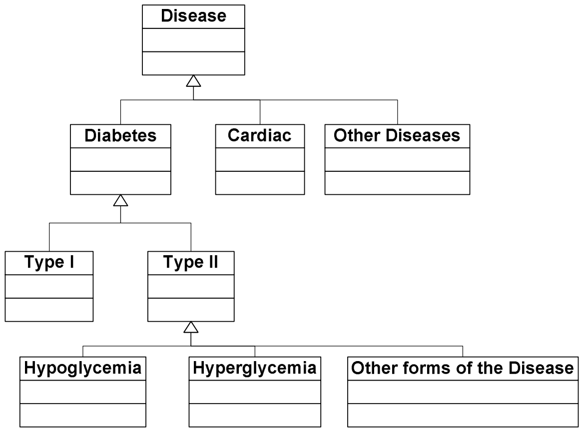

Figure 6. Disease Model

Each variant will specify what data type recordings should be made on the patient, e.g. blood sugar values, temperature, urination. It will specify threshold values for each data type and will also combine different thresholds to proactively detect negative trends in the patient's condition. The disease variant complete with all thresholds will be stored in the form of a policy, i.e. policy for each variant of a disease. These policies will reside in a Disease Repository and will be made accessible to related services. The policy for each variant will provide medical logic which can run as part of a service to aid in the medical care of acutely ill patients. This disease model will evolve to incorporate cross disease specifications, e.g. diabetic with pneumonia, with the input from medical specialists.

4. Case Study

Due to the considerable workload undertaken by a limited number of human resources within the hospital environment, load balancing and prioritization of emergency cases is of key importance to efficient healthcare systems. In this section we present a case study that will demonstrate the functionality of the group service communication supporting BANs. The case study will demonstrate how the group services are initially formed and dynamically reorganised. The focus of the case study will be for monitoring seriously ill patients in a hospital. The monitoring will consist of evaluating the data recorded by the body area sensors which are attached to the patient and using this information to detect negative trends in the patient's condition. Emergency situations can be detected and an appropriate response can be put in place in terms of staff reorganization and effective load balancing to handle the scenario.

There is one basic group which is fundamental to the acute health care system; this being a Patient Monitoring Group which is illustrated in Figure 7. This group includes the patient (fitted with body sensors), the nurse and doctor which are responsible for that specific patient, and the Hospital Management Service which holds the patients medical data. This group can use other services in order to help them achieve their tasks, i.e. provision of a detailed monitoring of a patient with proactive detection of negative trends.

The key identifier of the Patient Monitoring Group is the patient. There will be one instance of the Patient Monitoring Group per patient. The other members (e.g. nurse, doctor) can be replaced whenever necessary.

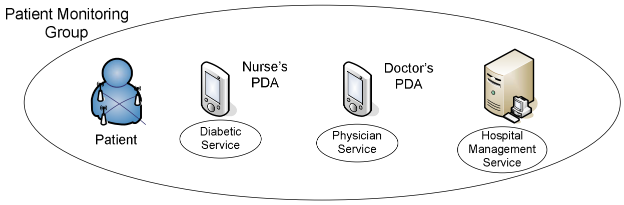

Figure 7. Patient Monitoring Group

Many other groups exist in the hospital environment, which includes:

- Body Area Net Sensor Group (BANS)
- Doctors Role Group – group of all available doctors employed by the hospital
- DoctorsOnSite Role Group – a group of doctors which are currently working in the hospital, i.e. possibly available for consultation. This is a subgroup to the doctors role group and contains the doctors details and their device details.
- Physician Service Group – a group which contains all the Physician Services, i.e. the services which run on the doctor's device.

These groups are used to form the communication groups which manage the patients.

4.1 Group Initialisation

Each patient monitoring group is initialized on patient admittance. The patient must be registered with the Hospital Mgt Service. The selection of members for the new group is driven by the configuration policy. This policy defines the mandatory members and what expertise they must have. An extract from a configuration policy is shown in Policy Extract 1. Figure 8 illustrates the interaction of the group initialization.

```
PolicyRule:
{CommonName ”GroupSetup”,
Events
createGroup,
Conditions
MandatoryMember1 – Is member of DocotorsOnSite  &&
Is Diabetologist && Is NotInSurgery
MandatoryMember2 – Is member of NursesOnSite &&
not high Patient Allocation
MandatoryMember3 - HospitalMgtService
Actions
initializeGroup}
```

Policy Extract 1. Configuration Policy

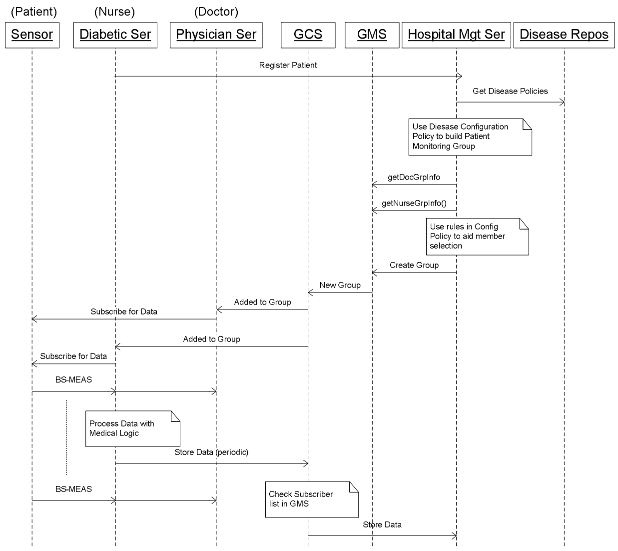

Figure 8. Group Initialization

Once the patient has been registered the Hospital Management Service will retrieve the appropriate policies from the Disease Repository. The configuration policy is then used to select the members of the new group. Details from other role and service groups are requested from GMS. The Hospital Management Service will select the members and create the group in GMS. As part of this creation the Diabetic Service on the Nurse's PDA will get a message indicating that it is part of a new group. It will then subscribe to the sensors for that specific patient. Once subscribed it will receive data from the BANS of the patient and it can apply the relevant group policy rules. Dependent on the configuration settings it will store the required data by sending it to the Hospital Management Service which in turn will update the medical records of the patient in the patient's profile.

4.2 Working Group Behaviour

The case study example identifies different scenarios around two independent patients. One patient is suffering from cardiac complications and has been an inpatient for a number of days and his condition has stabilized. The second is a diabetic patient who was recently admitted suffering from hypoglycemia. The two patient groups are illustrated in Figure 9.

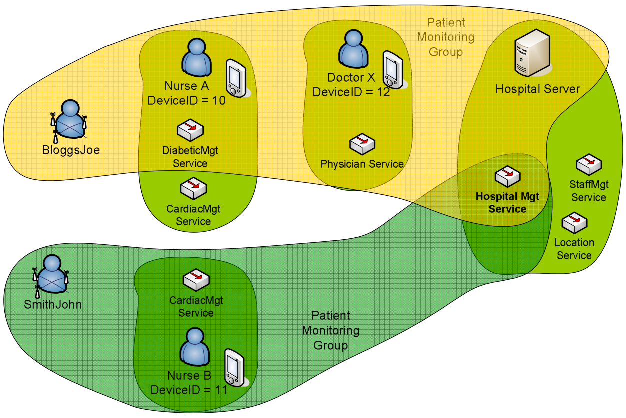

Figure 9. Graphical representation of Patient Monitoring Groups

The group which is monitoring the cardiac patient has no assigned doctor, this is the case because of the patient's stabilized condition and the nurse can manage the patient. Due to the instability of the diabetic patient the doctor has subscribed for regular blood sugar measurement reading to be sent to his device in order to closely monitor his condition.

Each patient monitoring group will have its own policy to determine the behaviour of the group and, the policy will be based on the disease which the patient suffers from. This policy will be decided upon on admission of the patient and downloaded and parsed from the Disease Repository. An extract of the Diabetic Policy is shown in Policy Extract 2.

```
PolicyEvent:
{Name "BSMeasurementEvent",
ParameterList "PatientIdentifier", "BS-Value"}

PolicyCondition:
{Name "HypoglycemiaHighPriority",
If BS-Value < 2mmol/l}

PolicyCondition:
{CommonName "HypoglycemiaLowPriority",
If (2mmol/1 < BS-Value < 3mmol/l)}

PolicyAction:
{CommonName "sendHighPriorityAlarm",
TriggerMsg "SendNotifToGroup"
Params "HIGH", "Blood sugar below 2mmol/l"}

PolicyActionAtomic :
{CommonName "sendLowPriorityAlarm",
TriggerMsg "SendNotifToGroup"
Params "LOW", "Blood sugar between 2-3mmol/l"}

PolicyRule:
{CommonName "RULE-BS-MEAS-HYPOGLYCEMIALOW",
Events
BSMeasurementEvent,
Conditions
HypoglycemiaLowPriority
Actions
sendLowPriorityAlarm}
```

Policy Extract 2. Diabetic Policy

The diabetic policy extract defines one *Event* for incoming blood sugar measurement values, a number of *Conditions* for hypoglycemia and two *Actions* which send different alarm types. The conditions are written in pseudocode form in order to aid readability, e.g. HypoglycemiaLowPriority condition is a range check for the incoming Blood Sugar measurement. One PolicyRule is specified which combines the different elements. It is triggered by the event, checks a condition and specifies the action to be triggered if the condition leads to true.

The difference in the patient's conditions directly affects the configuration of the Body Area Net Sensor Group. In case of the cardiac patient the current status is mainly stable and measurement values do not necessarily have to be available in real time. For power consumption issues it can be sufficient to locally store the data and regularly transmit the data based on a repeating time rule set in the Policy.

More critical is the setup for the Diabetes patient. Nevertheless, a simple exceedance of glucose levels does not always result in a situation the patient can not cope with on his own. But in case such an event is triggered the policy can activate further sensors in the BANS group to evaluate additional data (e.g. check for consciousness) and react appropriately.

This combination of local policies and business logic in cooperation with large scale group communication helps reducing communication overhead and prevent false alarms without decreasing measurement reliability.

4.3 Load Balancing & Prioritization

As mentioned earlier, due to the limited human resources in the hospital environment, load balancing and prioritization is of significant importance. Suppose the previously stable cardiac patient ECG readings deteriorate to dangerous levels, how will the medical resources be reorganized to prioritize emergency cases? The policy associated with that group will initiate the necessary reorganization. The extract in Policy Extract 3 specifies one PolicyRule which checks if the ECG values are high and if so subsequently verifies whether the receiving service is residing on the nurse's device.

```
PolicyEvent:
{CommonName "ECGMeasurement",
ParameterList "PatientIdentifier", "ECG-Value"}

PolicyCondition:
{CommonName "ECGHighPriority",
If ECG-Value is InDangerousRange}

PolicyCondition:
{CommonName "hasMemberDoctor",
If Member == Doctor}

PolicyCondition:
{CommonName "memberTypeNurse",
If Member == Nurse}

PolicyAction:
{CommonName "sendHighPriorityAlarm",
TriggerMsg "SendNotifToGroup"
Params "HIGH", "ECG values at dangerous levels"}

PolicyAction:
{CommonName "addMemberToGroup",
TriggerMsg "addMemberToGroup"
Params "{MemberType, Docotor}",
{BaseSelectionOn, {DoctorType,Cardiac} & {Load,
!Busy} & {Location,Closest}}

PolicyRule :
{CommonName "RULE-ECG-MEAS-ECGHIGH",
Events
```

```
        ECGMeasurementEvent
Conditions
        ECGHighPriority && memberTypeNurse
        && !(Policies.hasMemberDoctor)
Actions
        addMemberToGroup && sendHighPriorityAlarm}
```

Policy Extract 3. Cardiac Policy – Part 1

If these conditions are true then a final condition is checked to see if a doctor is part of the group, if this is untrue (the condition is negated) then two actions are triggered. The first action is to assign a doctor to the group and certain parameters are specified to aid selection, e.g. the doctor type must have cardiac expertise. The second action is to send an alarm to the group which highlights the current problem.

The flow of events is shown in Figure 10. Once the conditions are satisfied the Diabetic Service attempts to add the required doctor to the group. It queries GMS, a standard query which is available on all groups, to get information on all the members. The group queried is the DoctorsOnSite and for each member information is returned, e.g. doctors capabilities (diabetologist, cardiologist), doctors location (within the hospital), doctor's status with respect to load (may be mapped to the number of patient monitoring groups which they are included in). This data is used to select the most appropriate doctor. Once the Diabetic Service selects the doctor, GMS is requested to add this doctor's service to the group and then the second action of sending the alarm is executed by sending a message to the group via GCS.

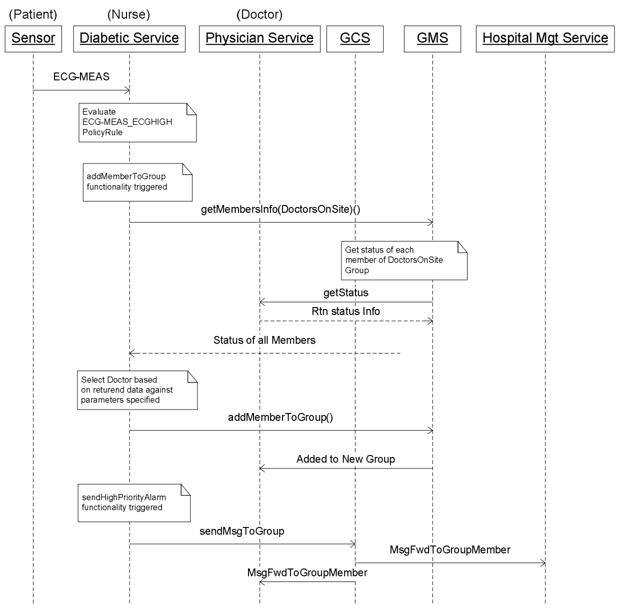

Figure 10. Group Reconfiguration Flow

Figure 10 illustrates the group reconfiguration flow, while Figure 11 illustrates diagrammatically the reorganization process from the pervious group formed in Figure 9. The Physician Service of the Doctor's device (ID=12) is now included in the SmithJohn's Monitoring Group. This highlights the dynamic reconfiguration of groups driven by the policies.

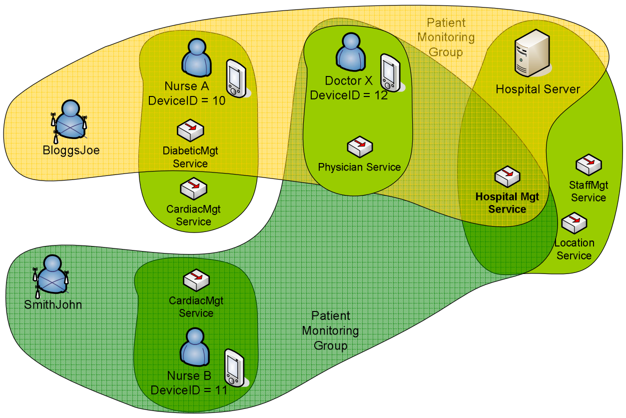

Figure 11. Group Reorganization after emergency

Once a doctor is assigned, a priority needs to be decided for the doctor with regards to data received from the diabetic and the cardiac patient. This can also be done through the policy. The extract in Policy Extract 4 has a PolicyRule defined which is triggered by the HighPriorityAlarm and has a condition based on the service on the doctor's device.

```
PolicyEvent:
{CommonName "HighPriorityAlarm",
ParameterList "PatientIdentifier", "HIGH", "ECG
values ????"}

PolicyCondition:
{CommonName "OtherGroupDataFlowTooHigh",
If DataStreams > 1}

PolicyCondition:
{CommonName "memberTypeDoctor",
If Member == Doctor}

PolicyAction:
{CommonName "dePrioritizeDataFlow",
TriggerMsg "dePrioritizeDataFlow"
Params "NonFrequent"}

PolicyAction:
{CommonName "subscribeForData",
TriggerMsg "subscribe"
Params "ECG Sensor"}

PolicyRule:
{CommonName "RULE-HIGHPRIORALM-DATAFLOW",
Events
        HighPriorityAlarm,
Conditions
        memberTypeDoctor &&
OtherGroupDataFlowTooHigh
Actions
        dePrioritizeDataFlow && subscribeForData}
```

Policy Extract 4. Cardiac Policy – Part 2

The second condition is related to the data flows which the service is receiving from other patient monitoring groups. If the service is receiving data streams from other groups then this condition is satisfied. The resulting action is to reduce the frequency of these other data flows from the other groups, so priority can be allocated to the cardiac issue. The second action is to subscribe to the ECG measurements of the cardiac patient which will provide immediate and frequent data for analysis.

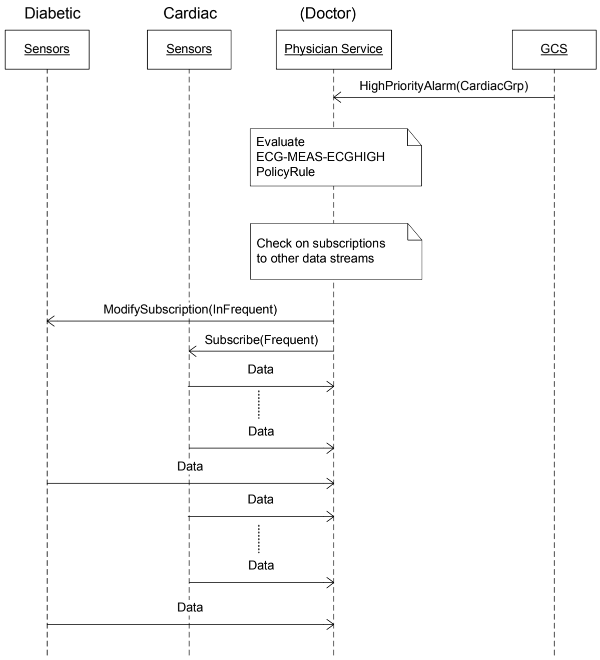

Figure 12. Prioritization between Groups

The data flow for the prioritization between groups is illustrated in Figure 12.

5. Conclusion

The use and interpretation of data which BANs provide is central to achieving pervasive healthcare. The architecture which eases this interpretation must be loosely coupled, adaptive and responsive. The MORE middleware is a toolset which can be used to achieve this and in particular the Group Service functionality which it provides. This functionality allows developers and medical officers to come together and build an adaptable healthcare system which will enhance patient care from the outset.

The design has been formulated and implementation is underway within the scope of the MORE project.

The constellation of groups of different sizes from local body nets up to large scale medical personal groups and the use of a policy based approach to control these groups is an integral part of the system. The derivation of the policies from the disease model is unique and it opens up the potential to inter disease solutions being addressed by technology.

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.

REFERENCES

[1] Alejandro Alonso, Daniel Berjon (Editors): “MORE Security Aspects”, Public Deliverable, EU-Project MORE, Madrid 2007, http://www.ist-more.org

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

[3] 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.

[4] 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

[5] R. Fisher, L. Ohno-Machado, D. Curtis, R. Greenes, T. Stair, J. Guttag, “SMART: Scalable Medical Alert and Response Technology”, Smart Medical Technologies Summit 2004, Houston, Texas, USA

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

[7] Tao Gu, Hung Keng Pung, Da Qing Zhang, “A service-orientated middleware for building context-aware services”, Journal of Network and Computer Applications 28 (2005) 1-18

[8] TMF (Ed: John Strassner): Shared Information/Data (SID) Model –Addendum 1-POL Common Business Entity Definitions – Policy GB922-1Pol, Ed. NGOSS R6.1, document Version 1.1, August, 2004

[9] Web Services Policy 1.5 Framework W3C Recommendation 04 September 2007 http://www.w3.org/TR/ws-policy/