BSN MIDDLEWARE
Abstracting Resources to Human Models

Pedro Brandão and Jean Bacon
Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, United Kingdom

Keywords: Body Sensors, Networks, System Architecture, Models.

Abstract: In the sensor network area, BSNs encompass a particular set of restrictions and conditions that separate them from normal WSNs. More so than WSNs, BSNs would profit from different types of sensing information and the sensor network itself provides more opportunities for different applications to use the same resources. However, the heterogeneity of sensor HW devices and the myriad of different applications that try to use them are an obstacle to its development. A problem is the need to address specific characteristics of the HW without abstractions that 1) provide the freedom to access the needed information while 2) complying to a set of requirements and 3) optimizing resource usage according to a set of metrics. We propose a middleware approach for abstracting lower level details from applications. We enrich the approach by building models in the middle layer fed with data from the sensor network and *query-able* from the application layer. Furthermore: a) applications should be able to set requirements to be met in providing the information, b) several applications should be able to share the same resources, c) the resources should be optimized so as to meet the requirements and prolong the lifetime of the BSN.

1 INTRODUCTION

Healthcare surveillance and fitness assessment has led to the proliferation of sensor use for human body monitoring purposes. This can be seen in commercial products for fitness assessment (eg.: (Vivometrics, 2007)) and research initiatives as (Lo and Yang, 2006; Korhonen et al., 2003) for health monitoring. Projects like Sesame (Computer Laboratory, 2007) at University of Cambridge that deal with assessing athletes performances, the Angel project (Garino et al., 2007) from the 6^{th} Framework EU program that focus on modelling, simulation and deployment of Body Sensor Network (BSN) applications and CodeBlue (Lorincz et al., 2004) from Harvard University that deals with network concerns in BSNs and Hardware (HW) for BSNs R&D) are examples of current research projects that focus on BSNs.

The trend and the need is here for BSNs, where the sensor network lies within the limits of the human body. There are several applications domains, be it monitoring vital signals of 1^{st} responders (eg.: (Vivometrics, 2007)) and/or victims (Lorincz et al., 2004) in disaster scenarios, health monitoring (Korhonen et al., 2003) (ranging from physical and chemical measures (Johannessen et al., 2005) to video imaging (Coimbra et al., 2006)) or specific disease treatment (eg.: diabetes (Steil et al., 2004)). Deriving user context and performance are also scenarios dealt with BSNs, for example: (Lukowicz et al., 2006) uses force sensors to detect muscle work and correlate it with fatigue and freshness of users to derive recognition patterns and in (Aylward and Paradiso, 2006) inertial sensors are used to discern group movement in dance.

It is worth noting that although the current momentum on BSNs, this research area can be dated back to 1961 with work from Mackay (Mackay, 1961) opening prospects for this trend. The area has been dormant until around a decade ago where interest in the subject reappeared. First as just a new view of Personal Area Networks (PANs) and gradually as a new type of network composed of sensor devices.

One issue that currently remains not addressed in BSNs is the possible heterogeneity of available information. Most of the references mentioned deal with just one type of information or have to go to great lengths to use different sources of sensing data. Being able to have plug-n-play sensors where adding a new sensor into the BSN would imply its availability as a source of information is, in our view, a major prob-

Brandão P. and Bacon J. (2009).
BSN MIDDLEWARE - Abstracting Resources to Human Models.
In *Proceedings of the International Conference on Health Informatics*, pages 245-250
DOI: 10.5220/0001545802450250

lem. This would enable a better and more accurate estimation of the current situation through the correlation of different sensing information. As an example, blood pressure is more accurately measured when taking blood pressure, blood flow and oxygen levels. By adding an intermediate layer (middleware) to the process we could have: **i)** application requests blood pressure information; **ii)** middleware retrieves data from the available sensors (which could involve activating them); **iii)** middleware aggregates the available information using a defined model, adding metadata about the information (confidence on the value (based on the model and the data available), error margin (taking into account statistics of the sensors used), time of assessment, etc); **iv)** middleware provides the application the information and metadata; **v)** application handles the information taking into account its metadata. The application request could (and most likely would) have requirements associated.

As such, we see the main problem in BSNs bound to the need to address specific characteristics of the HW without any abstraction that 1) provides the freedom to access the needed information while 2) complying to a set of requirements and 3) optimizing resource usage according to a set of metrics.

Our proposal is a middleware layer that abstracts the underlying BSN to the application, providing an information model to be queried. The information on the model itself is derived using the data provided by the BSN and applications state the models they want to use. Application requirements and resource optimization are also responsibilities of the middleware. Section 3 will detail this architecture.

1.1 WSN and BSN

BSNs have several similarities with Wireless Sensor Networks (WSNs), starting with being composed by sensor nodes (albeit different) that form a network. However we argue that there are several differences that lend BSNs prone to different problems, approaches and optimizations possibilities. Table 1 summarizes the aspects we regard fundamental, stating differences and similarities.

In our view, BSNs will have a central component that receives and acts upon the nodes as the applications using it deem necessary. A PDA, mobile phone or a more powerful node can be this component or can act as a Gateway (GW) to the central component (e.g.: a PC on a home environment). This central component (Base Station (BS)) will be able to control all the BSN, as the topology formed will be one-hop from it to the nodes, due to the close range. The nodes will refrain from much processing, delivering most of the data to the BSs[1]. The data set treated will be very heterogeneous, with possibly complex relationships among the different types of data. BSNs are prone to have different types of nodes added.

In WSNs the central component may or not exist and if present will have a more limited view of the whole system. WSNs tend to process the information in a distribute approach; with nodes treating, correlating and aggregating information, forming clusters, etc. The network in WSNs is multi-hop due to the usual large areas covered. The sensors in WSNs are homogeneous in terms of HW and data acquisition. As such, the information processed will normally be of the same type, although some frameworks do offer limited support for heterogeneity (see section 2).

Both frameworks have similar energy constraints (the research on recharging also shares some commonality (Paradiso, 2005) and (Roundy et al., 2004)).

The differences stated lead to an easier to manage network in BSNs in terms of connectivity establishment and coordination/optimization of sensor nodes. This opens more possibilities for optimizing the sensor network usage. However, BSNs monitor different types of information with complex interrelations between them, that can change from application to application. To add to this, applications will also have different requirements on the usage of the network (Quality of Service (QoS), error, rates, etc). A middleware should handle these concerns so to provide the applications with the abstract information they need releasing them from the specificities of the underlying resources.

2 RELATED WORK

In this section we will briefly describe some approaches that try to add flexibility and counteract heterogeneity. As an initial note, middleware specific for BSNs still remains largely unexplored, thus some of the work described here is from the WSN world.

As a counter example to the previous statement, we have the middleware from the Angel EU project (Garino et al., 2007) mentioned in section 1: Abstract Middleware Services (Fummi et al., 2007). The approach provides an environment aimed at early simulation and validation. It abstracts from the programmer the specific implementation to be used in the deployment. For this, it is able to be mapped to current existent middleware frameworks divided in: database abstraction, tuple-space based, object oriented and

[1]As counter example, the Equivital node (Equivital, 2007) processes all sensing information incorporating algorithms for producing the processed information.

Table 1: BSN vs WSN.

	BSN	WSN
Distrib.	i. Existence of a Base Station (BS); ii. BS collects, maintains and processes the data; iii. Nodes will do minimal processing, sending all data to the BS; iv. Centralized system where BS controls all nodes.	i. A BS may or not exist or there may be several BSs (may be mobile to collect info); ii. Same as in BSN, but also focus on on-demand querying; iii. Nodes will do processing, aggregation to alleviate communication or correlate results; iv. Distributed system where nodes decide cooperatively.
Comm.	i. One hop to BS; ii. Close range but attenuated by body.	i. Multi hop through network of sensor nodes; ii. Long range.
Energy	i. Constrained; ii. Replacement is difficult in in-body sensor nodes; iii. Recharging may be possible with scavenging or inductive coupling.	i. Constrained; ii. Replacement is difficult due to location, scale, etc; iii. Recharging may be possible in some scenarios with natural energy or scavenging.
Data	i. Correlation of different type of sensing information; ii. Likely to have new **types** of sensors added; iii. Prone to have different applications using the same resources.	i. Usually aggregating the same type of information; ii. Typically static in terms of node types (adding new sensors of the same type); iii. Usually deployed with one application in mind.

message based. The programmer is provided with an abstraction for the HW and network that enables the simulation of the application. The next phase of deployment implies letting go the network abstraction embedded in the middleware and having a separate model for simulating the network. The final phase maps the application to a specific implementation chosen. This approach does not free the application developer from the underlying resources characteristics per se, but provides a framework for early testing. The mapping to a specific framework carries the need to understand the framework and how it provides its results. Application requirements are said to be specified in the first phase of deployment, but they lack a description of the process, leading to the idea that the requirements are defined in the initial program by the developer in specific terms (taking into account the framework chosen).

Some research proposals for WSNs, tackle some of the issues we dubbed relevant. Namely: Middleware Linking Applications and Networks (MiLAN) (Heinzelman et al., 2004) tries to cope with *"the gap between the protocol and the application* [that] *is often too large to allow the protocols to be effectively used by application developers"*, by tackling what they devise as the features of sensor applications: distribution, dynamicity in the availability of sensor nodes, constraint application QoS demands, resource limitation (bandwidth and energy) and cooperative applications. This last issue relates to different applications using the same network to achieve different objectives. MiLAN tries to cope with different application requests using their QoS requirements as input. It also takes into account the network information (energy and bandwidth) and the system's information on the relevance of the different applications. MiLAN uses as input: i) the data (variables) that the application needs, ii) the required QoS for each variable and iii) the level of QoS that each sensor or group of sensors can provide for each variable. This information is defined through the use of two sets of graphs. These graphs enable the definition of sets of sensors to fulfil the requirements of the applications. Another component related to network characteristics defines another set of sensors capable of evaluating the needed information. The intersection of the sets gives the sensors such as to maximize the time that the information can be given (energy restriction).

In an attempt to ease application development, by providing a known development environment, some HW has been developed to directly support the Java programming language. Sun with SUNSpot (SUN, 2007) and Sentilla (Sentilla, 2008) are deploying sensor nodes that natively support Java. The rationale is to abstract from the application developer some of the underlying specifics of the HW by providing a familiar environment. However, the abstraction provided is at the language level, and most of the knowledge about the HW must still be known to correctly and efficiently implement the application. Our proposal intends to put the abstraction at the information level.

There are other abstractions providing querying interfaces that ease the development (such as TinyDB (Madden et al., 2005) and Cougar (Yao and Gehrke, 2002)). They handle some heterogeneity by taking the information sensed as input for Database (DB) like functions. However, this abstraction is limited in terms of functionality (DB like functions related to aggregation and threshold methodology) and does not provide the abstraction at the information layer. Other proposals like XMiddle (Mascolo et al., 2002) and Hood (Whitehouse et al., 2004) enable data sharing across different nodes relying on data structure abstractions (XML in XMiddle) to deal with the different data. Here the focus is on data replication and coherence and not explicitely on providing the notion of correlation between different types of information at a higher layer.

The work mentioned does not address the specificities of BSNs mentioned in section 1.1 (short range, BS availability, etc), which leaves room for further improvement. Also enabling the applications requirements to influence the resource allocation, thus crossing (or more precisely, adapting) layer information, will enable more opportunities for optimization (not needing to take worst case approaches).

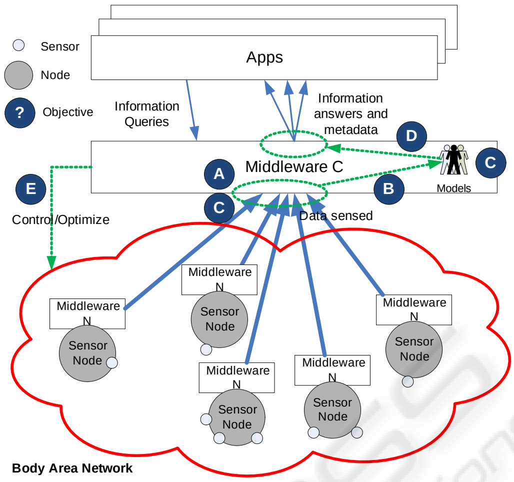

Figure 1: Architecture Proposed.

3 ABSTRACTION

As introduced in section 1 the motivation for this proposal lies in the lack of an information abstraction to the sensor resources that copes with: i) collection and aggregation of sensor data to provide higher level information; ii) Metadata about the information derived; iii) compliance to application requirements; iv) management and optimization of resources. The rationale being that, applications are not interested in handling the sensors but accessing specific information. This view is similarly defended for WSNs in (Römer, 2004). We introduce a middleware layer that provides the said abstraction and functionalities.

The general middleware objectives would be: **A)** collect data from sensor nodes; **B)** convert this data to relevant information in a human body model; **C)** collect metadata on the data received and correlate it to the information on the model; **D)** answer requests from applications based on the information in the model providing the related metadata; **E)** optimize resource usage (turn on/off, increase/decrease frequencies of data collection, etc) while complying to requirements set by the applications. Figure 1 depicts this architecture.

The next sections briefly describe the most relevant issues on this architecture.

3.1 Human Models

One of the interesting points in the problem at hand is related to the models to be used by the middleware. More specifically, how will the middleware map the data from the sensor network to the human model?

One approach is to develop a system similar to MiLAN (Heinzelman et al., 2004) (see section 2). It defines graphs that identify the relationship between sensed values and information assessment. The value can then be part of another graph that defines system states. The weights in these graphs describe the importance of the sensor data or value in the assessment of the information.

This system allows for a flexible definition of models. It nonetheless lacks some formality and standardization in terms of having a wider acceptance. The previous comment arises from the existence of projects that have a more wider and formal approach to define human biological models. As an example, the Physiome Project (Hunter et al., 2002) aims *“to use computational modelling to analyse integrative biological function in terms of underlying structure and molecular mechanisms”*. Here, the objective is to enable a mathematical handling of the Human body. To achieve this it follows a hierarchical modelling where the human body is divided in organs ⇒ composed of tissues ⇒ made up of cells ⇒ built on proteins ⇒ that are divided in atoms. The physical and chemical relationships between the hierarchies and the modules at each level are to be defined. The weight of introducing such a system in a middleware layer could be overwhelming or even an overkill. On the other hand, it can provide a more comprehensible (from the application point of view) model to query.

An architecture that enables applications to input the model that they want to use (stating how information is correlated) provides a flexible approach. These models should be nonetheless framed such that the middleware is able to: i) unequivocally identify the information and its source; ii) infer commonalities between different models as to optimize resources and iii) clearly interpret how the meta-data correlates in the model. Figure 2 gives a draft view of a possible tree model, where the difference from `BasicDataType` and `DerivedDataType` could be made irrelevant when the information that nodes produce is mapped to any level of the tree. Correlation of metadata information (error, rate, etc) is part of the model.

Defining a model that characterizes the mapping from sensor data to model information to serve as an input to the middleware is a major challenge.

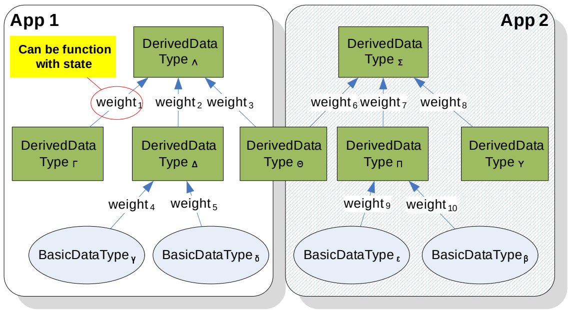

Figure 2: Models with detected common data type.

3.2 Middleware

As stated, the middleware will deal with managing resources so to fulfil applications requirements. As seen in figure 1 there will be parts running on the nodes (*Middleware N*) and the main block running on the central component[2] (*Middleware C*). The middleware on the node would be responsible for: **i)** advertising the nodes capabilities (sensing information, accuracy, rate, resources available (energy, processing, network), etc); **ii)** abide to the central component's commands (turn sensors on/off, change rate, etc); **iii)** answer information requests from central component. The central component, as mentioned, will have greater responsibilities so to: **i)** receive and incorporate models received from apps; **ii)** receive requests from apps with the associated requirements to be fulfilled; **iii)** derive from the different models, requirements and resources available a solution that satisfies the apps and optimizes resource usage; **iv)** control the sensors according to the solution found and request and receive information from them; **v)** maintain the model information and associated metadata from the sensor information (data and new sensor advertisements) so to re-evaluate the solution found.

Point iii) for the central component incorporates the optimization process and is the most relevant.

[2]The GW architecture is not discussed here, but would involve an extra block acting as a net GW.

3.2.1 Optimization

Finding a solution to maximize resource usage while satisfying application requests will mandate a definition of what are the requirements, metrics and controlled resources and how to map requirements to metrics so to optimize the said resources.

We have already undertaken work on the first point where requirements (eg.: real time, data updates frequency, etc), metrics (eg.: QoS related, quality of measurement related, etc) and resources (eg.: network architecture, processing power, etc) have already been drafted. Optimization algorithms will play the relevant role in this issue.

Resources in these networks may change very abrupt and unpredictably. This raises the problem of how does the middleware handle the commitment to fulfil requirements when resources are very unstable. A simple first solution is the usage of metadata indicating the reliability of the information and triggers when resources disappear, letting the application deal with the failure. However, this should only occur when no other solution is feasible from the central middleware's perspective.

The issues described in section 1.1 that differentiate BSNs from WSNs will take great influence on the middleware. As stated, management will be simpler with a central point and one hop communication paths while the range of application and requirements will pose severe constrains and opportunities on the optimization process. This is then coupled with the need to merge different models from different applications to reach the best solution.

4 CONCLUSIONS

Our proposal aims to release the BSN application developer from the details of the underlying HW by providing an abstraction to these resources, enabling the application to access the underlying information correlated and aggregated from the data provided by the BSN. Thus, applications can gain with data from different types of sensors (for the same or different types of information) without needing to address their specificities; applications can set requirements to be met in providing the requested information; several applications can more easily be accommodated on top of the same set of resources; resource usage can be optimized while taking into account the previous mentioned points; sensor plug-n-play capability can more easily be attained; compartmentalization of responsibilities provides better room for improvements/updates (eg.: updating resource optimizations algorithms will not imply changes in applications). Applications can access higher level information based on a model that derives its values from the data on the sensor network.

Models, the mapping between sensor data and model information and the optimization of resources while meeting requirements are the main challenges. Other issues like network topologies and service discovery are also future work as the former will be related to resources and the latter with the ability to access information in a plug-n-play architecture.

Experiments (healthcare and physical assessment) will be undertaken at a future point to test the proposed architecture.

ACKNOWLEDGEMENTS

Pedro Brandão is partially funded by Fundação para Ciência e Tecnologia from Portugal.

REFERENCES

Aylward, R. and Paradiso, J. A. (2006). Sensemble: A wireless, compact, multi-user sensor system for interactive dance. In *NIME*, pages 134–139, 1, place Igor-Stravinsky, 75004 Paris. IRCAM.

Coimbra, M., Campos, P., and Cunha, J. P. (2006). Topographic segmentation and transit time estimation for endoscopic capsule exams. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages 1164–1167.

Computer Laboratory, D. T. G. (2007). *Sesame Project*. www.sesame.ucl.ac.uk.

Equivital (2007). *Equivital*. www.equivital.co.uk.

Fummi, F., Perbellini, G., Pietrangeli, R., and Quaglia, D. (2007). A middleware-centric design flow for networked embedded systems. In *Design, Automation & Test in Europe Conference & Exhibition*, pages 1048–1053, Nice, France. EDA Consortium.

Garino, P., Gorski, J., Gouw, M. D., Huebner, A., Quaglia, D., and Willig, A. (2007). An integrated approach for the design of applications based on body sensor networks to improve quality of life. White paper, Angel Project, Angel project white paper.

Heinzelman, W. B., Murphy, A. L., Carvalho, H. S., and Perillo, M. A. (2004). Middleware to support sensor network applications. *Network, IEEE*, 18:6–14.

Hunter, P., Robbins, P., and Noble, D. (2002). The IUPS human physiome project. *Pflugers Arch*, 445:1–9.

Johannessen, E. A., Wang, L., Reid, S. W. J., Cumming, D. R. S., and Cooper, J. M. (2005). Implementation of radiotelemetry in a lab-in-a-pill format. *Lab on a Chip*, 6:39 – 45.

Korhonen, I., Parkka, J., and Gils, M. V. (2003). Health monitoring in the home of the future. *Engineering in Medicine and Biology Magazine, IEEE*, 22:66–73.

Lo, B. P. and Yang, G.-Z. (2006). Body sensor networks: Infrastructure for life science sensing research. In *Life Science Systems and Applications Workshop, 2006. IEEE/NLM*, pages 1–2.

Lorincz, K., Malan, D. J., Fulford-Jones, T. R., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., and Moulton, S. (2004). Sensor networks for emergency response: challenges and opportunities. *Pervasive Computing, IEEE*, 3:16–23.

Lukowicz, P., Hanser, F., Szubski, C., and Schobersberger, W. (2006). Detecting and interpreting muscle activity with wearable force sensors. In *Pervasive Computing*, volume 3968/2006 of *Lecture Notes in Computer Science*, pages 101–116. Springer Berlin / Heidelberg.

Mackay, R. S. (1961). Radio telemetering from within the body. *Science*, 134:1410.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst*, 30:122–173.

Mascolo, C., Capra, L., Zachariadis, S., and Emmerich, W. (2002). Xmiddle: A data-sharing middleware for mobile computing. *Wireless Personal Communications*, 21:77–103.

Paradiso, J. A. (2005). Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*, 4:18–27.

Römer, K. (2004). Programming paradigms and middleware for sensor networks. *GI/ITG Fachgespräch Sensornetze, Karlsruhe*.

Roundy, S., Wright, P. K., and Rabaey, J. M. (2004). *Energy Scavenging for Wireless Sensor Networks: With Special Focus on Vibrations*. Kluwer Academic Publishers.

Sentilla (2008). *Sentilla*. www.sentilla.com.

Steil, G. M., Panteleon, A. E., and Rebrin, K. (2004). Closed-loop insulin delivery-the path to physiological glucose control. *Adv Drug Deliv Rev*, 56:125–144.

SUN (2007). *SunSPOT*. www.sunspotworld.com.

Vivometrics (2007). *VivoMetrics*. www.vivometrics.com.

Whitehouse, K., Sharp, C., Brewer, E., and Culler, D. (2004). Hood: a neighborhood abstraction for sensor networks. In *International Conference On Mobile Systems, Applications And Services*, pages 99–110, Boston, MA, USA. ACM.

Yao, Y. and Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, 31:9–18.