

# Programming Body Sensor Networks

**Talles Marcelo Gonçalves de Andrade Barbosa**

*Catholic University of Goiás, Brazil*

*University of Brasília, Brazil*

**Hervaldo Sampaio Carvalho**

*University of Brasília, Brazil*

**Iwens Gervásio Sene Jr.**

*Catholic University of Goiás, Brazil*

*University of Brasília, Brazil*

**Francisco A. de O. Nascimento**

*University of Brasília, Brazil*

**João Luiz Azevedo de Carvalho**

*University of Southern California, USA*

**Adson Ferreira da Rocha**

*University of Brasília, Brazil*

## INTRODUCTION

The miniaturization and cost reduction of microelectronic devices have been leading to the development of new technologies. Wireless sensor networks (WSNs) are one example of these new technologies. A WSN is a distributed system that is composed of autonomous units with sensing capabilities (sensor nodes), interconnected by wireless communication. WSNs have been successfully applied in the monitoring of the human body. This development led to a new concept: Body Sensor Networks (BSNs).

According to Yang (2006, pp. 5), BSNs have specific requisites, when compared to WSNs. Among these requisites, we highlight the following:

- High level of security, in order to guarantee the confidentiality of information;
- Biocompatibility and biodegradability;
- Higher sensitivity to data loss, and the consequent need for mechanisms to ensure a minimum Quality of Service (QoS);
- Need for context awareness, as the physiological variations are strongly related to the changes in the context in which the user is in;

- Low number of sensor nodes, which should, however, be usually more precise than sensors for other applications of WSNs;
- Requirement of nodes with the capability of running multiple tasks.

Pervasive monitoring demands great adaptation capability from the BSN. Moreover, cases in which the decision made by the system can be different from the decision that would be made by a healthcare professional are frequent. Therefore, besides intelligent algorithms that allow autonomous operation, BSNs need mechanisms that allow changes in their behavior in order to become a clinically useful tool. According to Baldus, Klabunde, and Müsch (2004), “the BSN has to work automatically, but has also to be always under explicit control of any clinician.”

Thus, specific models that include programmability as a functional requisite are important in a software architecture designed for BSNs. However, the greatest challenge is to allow the modifications to be made not only to the structure of the software, but also in its behavior, without excluding the capability for autonomous operation of the system.

According to Barbosa, Sene, Carvalho, da Rocha, Nascimento, and Camapum (2006), two important

concepts related to programmability in BSNs are: (i) deployment-time programmability, and (ii) run-time set-up. The deployment-time programmability refers to the definition of software artifacts and algorithms that are embedded in the sensor node. In BSNs, the inclusion of this functionality requires a programming interface that is suitable for healthcare personnel, as well as intelligent compilers. Intelligent compilers should be capable of handling implicit functional and nonfunctional requisites of a program. As an example, the inclusion of mechanisms and policies for energy saving could be treated by these structures.

The run-time set-up refers to the capability for adjustments in run-time. The BSN should provide interactivity between the healthcare professional (the BSN manager) and the system. As a requisite, sensor nodes need mechanisms that allow a better control of the tasks that are being run. A possible solution is the use of data structures that allow preemptive multitasking.

The goal of this article is to present the current state of the art, regarding programmability in BSNs. Moreover, we want to present potential benefits of a paradigm shift in which healthcare professionals become the actual programmers and maintainers of the BSNs. With that in mind, we briefly present a software architecture that has been developed with the goal of allowing programmability at network and sensor node levels.

## BACKGROUND

Many issues related to software for BSNs have not been discussed yet. Among them, we can mention: (i) the development of graphical user interfaces directed to healthcare personnel; (ii) the hardware abstraction layers (HALs); (iii) the standardization of services and information structuring (BSN ontology); and (iv) programmability at sensor and sensor node levels. The solutions to these problems can lead to improvement of the effectiveness of these systems.

Currently, the software used in BSNs has the following characteristics:

- They are composed of proprietary systems built based on a specific architecture (hardware), and designed for handling a single application. They have little or no modularization, and they are not committed to software development methodolo-

gies. In general, they do not employ multiprogramming. Sensor nodes are usually viewed just as sources of data, and all processing is performed in a gateway—an element that interconnects the BSN to other systems—or in a Local Processing Unit (LPU), which is an element to where the data are transmitted. Some examples of such systems are presented in Asada, Shaltis, Reiner, Sokwoo, and Hutchinson (2003), Valdastrì, Menciassi, Arena, Caccamo, and Dario (2004), Linz, Kallmayer, Aschenbrenner, and Reichl (2006), Kara, Kemaloglu, and Kirbas (2006), and Chakravorty (2006).

- These systems are usually based on a generic, general purpose model. It is usually based on the NesC programming language (Gay, Levis, von Behren, Welsh, Brewer, & Culler, 2003), on the TinyOS Operating system (Hill, 2003), and on a network programming system, the Deluge (Hui & Culler, 2004). All of these systems are free of charge, and they were developed by the University of California, in Berkeley. CodeBlue (Welsh, 2006), WHMS (Jovanov, 2006), and UbiMon (ICL, 2006) are examples of designs that use the TinyOS framework. Other examples are presented in (Bauldus et al., 2004) and in (Farshchi, Nuyujukian, Pesterev, Mody, & Judy, 2006).

Regarding programmability, systems built based on TinyOS have the following limitations when applied to BSNs:

- The NesC programming language imposes a peculiar syntax, based on concepts emerged from software engineering. Without the knowledge of programming logic and the expertise in managing software components, it is virtually impossible for a nonspecialized user to use this system.
- The multiprogramming model used in TinyOS is not interactive enough. It offers little control over the activities (tasks) run by the sensor node, because there is no context switch. Tasks cannot be immediately interrupted or replaced in order to answer a policy established by the application, or to answer to a command issued by the user. Moreover, according to Han, Bhatti, Carlson, Dai, Deng, Rose, Sheth, Shucker, Gruenwald,

8 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/programming-body-sensor-networks/13052](http://www.igi-global.com/chapter/programming-body-sensor-networks/13052)

## Related Content

---

### Assessing HIPAA Compliance of Open Source Electronic Health Record Applications

Hossain Shahriar, Hisham M. Haddad and Maryam Farhadi (2022). *Research Anthology on Securing Medical Systems and Records* (pp. 995-1011).

[www.irma-international.org/chapter/assessing-hipaa-compliance-of-open-source-electronic-health-record-applications/309039](http://www.irma-international.org/chapter/assessing-hipaa-compliance-of-open-source-electronic-health-record-applications/309039)

### Doing and Understanding: Use of Case Studies for Health Informatics Education and Training

Cynthia LeRouge, Herman Tolentino, Sherrilynne Fuller and Allison Tuma (2013). *Cases on Healthcare Information Technology for Patient Care Management* (pp. 1-34).

[www.irma-international.org/chapter/doing-understanding-use-case-studies/73939](http://www.irma-international.org/chapter/doing-understanding-use-case-studies/73939)

### A Comparison of How Canada, England, and Denmark are Managing their Electronic Health Record Journeys

Denis Protti (2010). *Health Information Systems: Concepts, Methodologies, Tools, and Applications* (pp. 402-417).

[www.irma-international.org/chapter/comparison-canada-england-denmark-managing/49877](http://www.irma-international.org/chapter/comparison-canada-england-denmark-managing/49877)

### Resource Alignment of ICT in Taiwanese Small Dialysis Centers

Yu-An Huang and Chad Lin (2010). *Handbook of Research on Advances in Health Informatics and Electronic Healthcare Applications: Global Adoption and Impact of Information Communication Technologies* (pp. 221-233).

[www.irma-international.org/chapter/resource-alignment-ict-taiwanese-small/36384](http://www.irma-international.org/chapter/resource-alignment-ict-taiwanese-small/36384)

### Characterization of HRV by Poincare Plot Analysis among the Female Tea Garden Workers of Northern Hilly Regions of West Bengal

Somsirsa Chatterjee, Ankur Ganguly and Saugat Bhattacharya (2010). *International Journal of Healthcare Information Systems and Informatics* (pp. 49-59).

[www.irma-international.org/article/characterization-hrv-poincare-plot-analysis/42997](http://www.irma-international.org/article/characterization-hrv-poincare-plot-analysis/42997)