Gait Analysis Using Virtual Body Sensor Networks for Biomedical Applications

Parthasarathy V, Sharmila P, and Hemalatha S
Vel Tech Multitech, Chennai, India
E-mail: parthasarathy@veltechmultitech.org, sharmila@veltechmultitech.org

Abstract—In BSNs Signal processing usually consists of multiple levels of data abstraction, with raw sensor data to data which is calculated from each processing steps that includes feature extraction and classification. Here we present a multi-layer task model based on the concept of Virtual Sensors in order to improve design reusability and architecture modularity. Virtual sensor networks (VSNs) is an emerging form of collaborative wireless sensor networks. Virtual Sensors are abstractions of components of BSN systems that involve sensor sampling and tasks processing and issues data upon external requests. The model of virtual sensor implementation depends on SPINE2, which is an open source domain-specific framework that is developed to support distributed sensing operations and processing of signal for wireless sensor networks and enables code efficiency, reusability and application interoperability. This proposed model is applied in the framework of gait analysis through wearable sensors. According to SPINE2-basedVirtual Sensor architecture a gait analysis system is developed and it is experimentally evaluated. The results obtained confirm that great value can be achieved to design and implement BSN applications through the Virtual Sensor approach at the same time maintaining high efficiency and accuracy.

***Index Terms*—Body Sensor Networks, Virtual Sensors, SPINE2, Signal Processing**

I. Introduction

Body Sensor Networks (BSNs) is a wireless network of wearable computing devices and it signifies a new application domain have been receiving a rapid development recently due to its potential benefits for health-care, entertainment, sport, fitness and social interaction [1]. BSNs has several properties that is not present in standard WSN deployments, that includes single-hop networking, which is a more powerful node known as base station and applications that requires an extensive signal processing and pattern recognition. Ref. [2] Among the domain-specific frameworks the SPINE framework, has unique features in terms of efficiency, effectiveness and usability which allows for a more rapid prototyping of efficient solutions of signal processing on the sensor nodes of BSN. Specifically, the SPINE framework allows the implementation of multiple signal processing functions, either on top of each other or by running independently on top of the sampled sensor data. In addition, it also facilitates selective activation of these functions at run-time that offers the flexibility to vary the node side complexity. Ref. [3] Though frameworks such as SPINE provide us effective and efficient abstractions for BSN application development and exploitation of the concept of virtual sensor that is recently introduced also in the context of WSNs may further enhance the modeling effectiveness of the solutions that is provided. In this paper, we present a task abstraction layer conceived for BSNs, called BSN-oriented Virtual Sensors. (BVS), that allows for multiple stages of processing. Ref. [4] it describes the implementation of BVS through SPINE2 .This is a new version of SPINE based on a task-oriented model. Particularly, the application of BVS is explained by developing a module based on Hidden Markov Model (HMM) to extract temporal parameters from gait with the final goal is to describe a Gait Virtual Sensor that is useful for real-time postural analysis of assisted livings. The rest of this context is organized as follows. In Section 3, Multilayer signal Processing for virtual sensor is discussed. The results and discussions are presented in Section 4 and Section 5 concludes the work with inferences.

Most of the applications of wireless sensor networks due to the constraints of resources makes difficult to design an operating system that is both powerful and lightweight. Because of resource abundance, traditional environments and desktop operating systems provide APIs for many different issues and domains. On the other hand, WSNs operating systems, such as TinyOS [2], gives only minimal facilities, that includes communication and scheduling of process. Those problems which allocate common structure, the abstraction given by a software framework can increase developer productivity and use domain or problem- -specific optimizations. Structures for conventional wireless sensor networks includes TinyLIME [3], (Global Sensor Network) GSN [4], and (Abstract Task Graph) ATAG [5]. Authors in [4]-[6] these works propose generic abstractions for all system components, which provide us with easier integration of new functionality and components on the system. However, certain application domains are not so effective in providing efficient and optimized support to specific tasks at communication, sensor and processing levels [7]. In fact,

Manuscript received January 26, 2015; revised March 31, 2015.

doi: 10.18178/ijpmbs.4.2.123-127

they aim to program general-purpose WSN applications and not a domain-specific WSN applications.

II. Related Work

In Ref. [6], authors describe Virtual Sensors as software sensors that allows indirect measurements of abstract conditions by means of combining sensed data from a group of heterogeneous physical sensors. In specific, a virtual sensor is defined by four parameters: (i) input data types that are physical (low-level)data types to compute the desired abstract measurement; (ii) aggregator, that is a generic function defined to operate over the specific (possibly heterogeneous) input data types to calculate the desired measurement; (iii) resulting data type, which is the abstract measurement type that is a result of the aggregation; (iv) aggregation frequency, such as the frequency in which this aggregation has to be made and this frequency determines how steady the aggregated value is with actual conditions. Authors also presented a middleware which is used for programming WSN applications designed around the defined virtual sensor abstraction. In particular, the middleware that is implemented in nesC on TinyOS, provides an API to program virtual sensors, services for sensor discovery and communication.

Although the proposed approach is suits for general-purpose WSN applications, it should be customized to support the characteristic abstractions of BSN applications. Truely, the approach does not consider fine-grained, signal-processing oriented virtual sensors and it is mainly based on query driven virtual sensors instead of data- driven virtual sensors that would have greater performance in BSNs. A framework for building virtual sensors and actuators in wireless sensors is presented. In particular, authors propose virtual nodes as a programming Abstraction that simplifies the development of decentralized applications of WSN. The data gathered by a set of sensors can collected, processed in accordance to an application-provided aggregation function, and it is then perceived as single virtual sensor reading. Alternatively, a virtual actuator provides a single entry point to distribute the commands to a set of real actuator nodes. Using logical neighborhoods the set of physical nodes to be abstracted into a virtual one is specified. A virtual sensor is specified as a set of inputs from neighbor sensors or virtual sensors and an aggregation function processing the received inputs and producing an output. Using a high-level programming language (an extension of SPIDEY) Virtual sensor and actuators are programmed and then it is translated into nesC/TinyOS through the SPIDEY translator purposely extended. Even though the approach allows for hierarchical composition of virtual sensors, it is best suited for environmental monitoring or building automation rather than for BSN because it does not provide BSN-oriented signal processing intensive abstractions and optimized fine-grain virtual sensors. The Virtual Sensor Networks concepts (VSNs) has also been described in multiple works. In [8], author's present VIP Bridge that aims to connect heterogeneous sensor networks with IP based wired/wireless networks and it integrate these sensor networks into one VSN. This allows discovering sensor nodes located in different and heterogeneous sensor networks. An agent-based approach for VSNs is proposed. Generally Agents support VSN connectivity, coverage, membership, formation as well as power management. Such proposals can be customized to enable the integration of BSNs to have networks of BSN on which Virtual Sensors can be created for monitoring not only a given person but also an even large group of people wearing heterogeneous BSNs and, notably, their gesture-initiated interactions.

III. Experiments

A virtual sensor based system is implemented as a test application for virtual sensors in SPINE2 which is based on a Hidden Markov Model (HMM).Each data sample is associated with a state in HMM. There are four parts in HMM annotation system such as sampling the acceleration data, pre filtering, feature extraction from filtered data, HMM based annotation of events. The SPINE coordinator generates a request which initiates the configuration of each virtual sensor. When all the virtual sensors are initialized, the HMM virtual sensor starts to produce the output. The accelerometer defines the sampling interval. Based on the actual complexity the each virtual sensor can be implemented through one or more SPINE2 tasks.

A. BSN-Oriented Virtual Sensors

Physical quantities such as temperature, acceleration, or sound are mapped by the Physical sensors onto a data value and produce an output. This output is generated when inputs change, as the result of an event, or in response to a request. Generally Physical sensors are transducers which convert value from one form to another using physical process. The observed similarity is the inspiration behind the virtual sensor abstraction. Each processing task can be represented as a virtual sensor. Thus, if we consider a complete BSN system, we can model its data processing part as a multi-level hierarchy of virtual sensors. In addition, virtual sensors may be either implemented directly in a programming language, or as networks of existing virtual sensors. A user requests certain outputs for the given specified inputs. The Virtual Sensor Manager handles the request that configures a set of virtual sensors to handle the computational task. Buffer Manager is used by the virtual sensors to setup communication through the use of efficient buffers. The system gets activated, once configured and well cooperated by virtual sensors to produce the final outputs.

A new level of abstraction is provided by the virtual sensors at the software level by allowing signal processing tasks to be defined and composed easily. In addition to this, VS abstractions allow signal processing tasks to be modified or changed at design or runtime without affecting the rest of the system. A processing task applied is represented by every component, to a stream of data originated from physical sensors and can

be modelled as a virtual sensor. A set of input and its configuration defines the output of each virtual sensor. Formally virtual sensor is defined as

$$VS_i = \{ I_i, O_i , C_i\} \quad (1)$$

where, C_i denotes the configuration, I_i denotes the set of inputs, O_i denotes the set of outputs

Each virtual sensor configuration defines the type of its inputs and outputs, the particular implementation used for a given computational task and parameters required for implementation. Re-initialization could happen at any time because the current configuration of a virtual sensor may be invalidated by changes in its inputs or connections with other virtual sensors. To configure/re-configure the system at run time, the VSM manages a table that maps each available combination of possible inputs and outputs to the desired virtual sensor implementation. This can be represented by the set. Each entry is defined as follows:

$$a = \{ t_{in} , t_{out} , \mu\} \quad (2)$$

where, μ is a particular virtual sensor implementation.

Reconfiguration can alter parameters of a given implementation, if the modification is not drastic enough to require changing the virtual sensor implementation. VSM includes the address of the selected virtual sensor implementation and the required configuration parameters, during the configuration of a virtual sensor. Signal processing for BSNs frequently relies on data combination from multiple sources and locations. Therefore, virtual sensors can have multiple inputs from different sensor nodes. Virtual sensors implicitly use buffers for communications as to avoid synchronization issues. The dynamic buffer allocation and managing the system data flow is controlled by Buffer Manager (BM). The BM checks whether the buffer has enough information for any of the readers, each and every time the producer writes to the buffer and signals them when they can access the data.

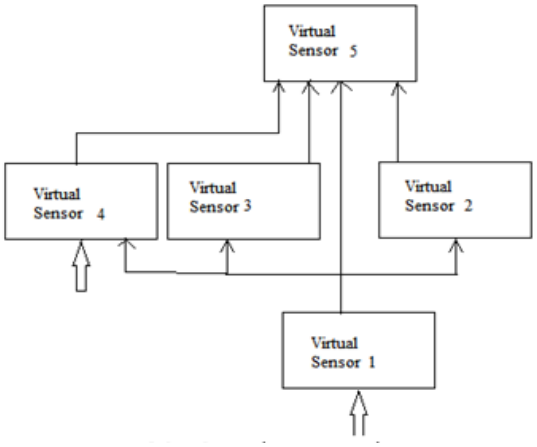

Figure 1. Dependent input and output.

B. Overall System Configuration

Sensors do not hold any information about other virtual sensors, while the overall system relies on their cooperation. The VSM receives the VS topology at the beginning of the system execution, configuration graph [9]. Based on the topology configuration requirements, the VSM initializes the appropriate VSs and connects them as required. Input and output types are a property of each virtual sensor. A virtual sensor output can also be an input of another virtual sensor. For example, in Fig. 1, configuration of $VS_3$and VS_2 depends on the input they receive from VS_1.

The VSM initializes VSs in a specific order, to simplify the configuration and re-configuration process requirement that each virtual sensor cannot be created until all inputs are configured. With a topographical sort of the topology configuration graph, this ordering can be determined.

IV. Results and Discussion

Ref. [5] for the development of signal processing applications on WSNs a novel framework called SPINE2 was specifically designed which is based on task oriented paradigm. The SPINE2 task-oriented application description is shown in the Fig. 1.The interconnected tasks forms a direct graph representing chains of operations that include sensor data acquisition, processing and merged result transmission. As the framework manages the deployment phase of a distributed application, each user can decide which of the wireless sensor network each task is allocated on. The SPINE2 task-oriented language composed of three main categories: *data-processing*, *data-routing* and *time-driven tasks*. Ref. [5] The first category makes it available functions related to data processing and archiving and the second one provides store-and-forward and data replication functionalities, whereas the latter provides a mechanism for the time-driven tasks. Each task is coupled with a description that consists of a set of configuration parameters which defines the behavioral characteristics of a given instance of a task type. Different types of tasks are handled in Spine2 such as: timing task, sensing task, processing task, transmission task, storing task, loading task, and split task, merge task and Historical merge task.

Ref. [10], [11], a novel architecture of optical router is designed using RTL and in order to use the bulk flow TCP it is implemented in. Various functional blocks of optical router like fabrication of flow from the incoming packets, processing the flow for routing process, contention resolution, buffering the packet or flow based on the wavelength and channel availability and transmission of the flow for destination have been successfully designed in RTL and implemented in FPGA. The numbers of slices utilized by various flow stages have also been simulated. The proposed optical router is tested for its performance in terms of frequency of operation and the area of utilization by various functional units. We have also implemented the central control unit for this optical route

Implementation of each virtual sensor can be done by using SPINE2 tasks Fig. 2. Initially the implementation was started with MATLAB. To match the sensor nodes

capabilities the code was simplified. Then the C code was adapted to SPINE2 and it was ported to nesC in order to perform an adhoc implementation of sensor nodes. Freezing of the system was the major problem which is because of higher processing overhead. Ref. [6] This caused communication interruptions. Considering these problems code optimization and corrections on the respective components was done. From the result depicted in Fig. 3 and Fig. 4 it is observed that there is a drastic difference between the sensor node output and rest of the others. The transition states were not synchronized completely. The total time for the complete sequence transitions was almost same for the entire experiment.

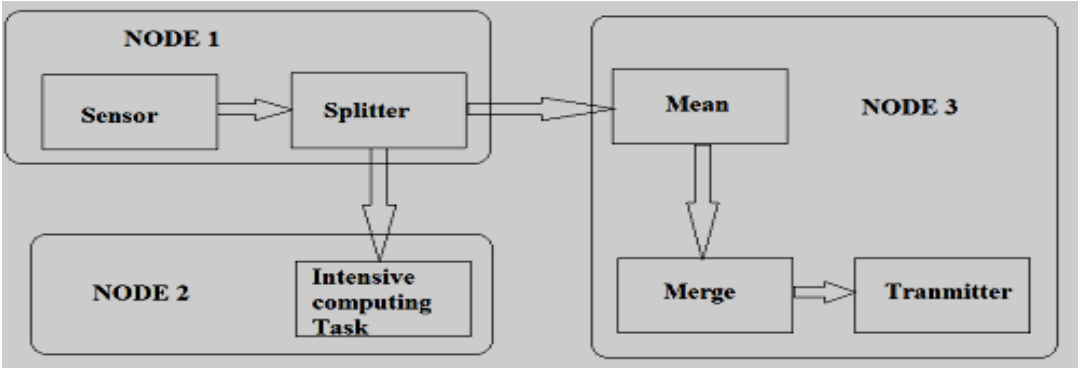

Figure 2. SPINE2 task – oriented distributed application.

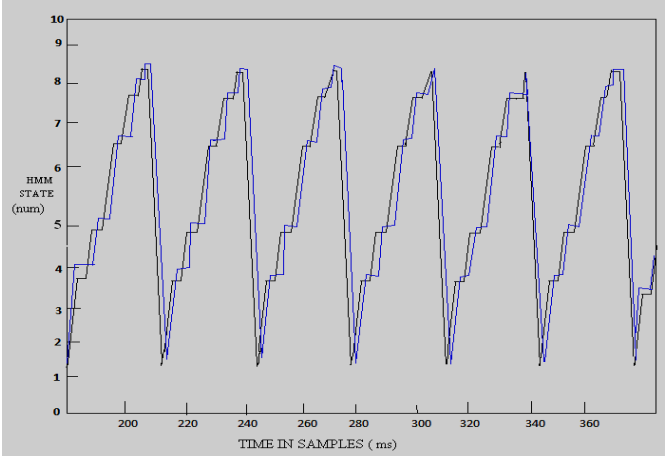

Figure 3. Implementation of virtual sensors.

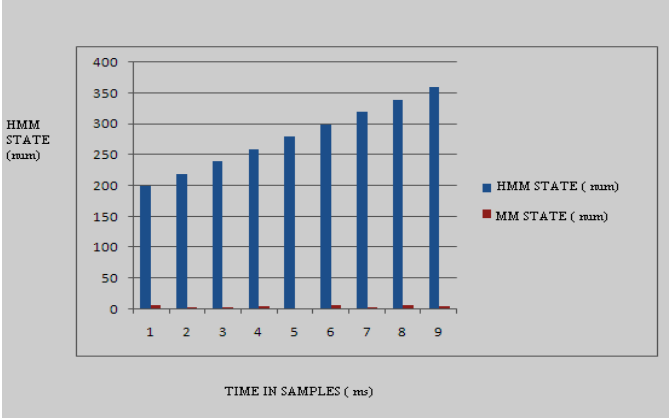

Figure 4. Virtual sensors in SPINE2.

V. Conclusion

This paper defines a new approach to design of BSN applications which is based on the virtual sensors concepts and it relies on the SPINE2 framework. In the BSN applications there is need of new abstractions that could capture the successive stages of the data processing and classification, which includes sensor data fusion.

Virtual sensors allow abstracting even complex systems that behave like sensors, which favors code modularity and reuse. If a programmer needs to update an implementation of a virtual sensor or changing environmental conditions require that a different implementation strategy is adopted, it is more sufficient to replace that particular portion of code without changing the rest of the system implementation. Also virtual sensors provides us simple coding scheme in which network abstraction, local sensors, remote sensors can be treated in a same fashion.

ACKNOWLEDGEMENT

Authors deliver their gratitude to FIST-DST Program (No. SR/FST/College-189/2013) Govt. of India for laboratory facilities support

REFERENCES

[1] F. Bellifemine, G. Fortino, R. Giannatonio, R. Gravina, A. Guerrieri, and M. Sgroi, "Spine: A domain-specific framework for rapid proto-typing of wbsn applications," *Software: Practice and Experience*, Sep 7, 2010.

[2] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstracting data from physical sensors," in *Proc. Int. Symp. World of Wireless, Mobile and Multimedia Networks*, 2006, pp. 587-592.

[3] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. Murphy, G. Picco, *et al*., "TinyLIME: Bridging mobile and sensor networks through middleware," in *Proc. 3rd IEEE Int. Conf. Pervasive*, 2005, pp. 61-62.

[4] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proc. 32nd Int. Conf. VeryLarge Data Bases. VLDB Endowment*, 2006, pp. 1199-1202.

[5] A. Bakshi, V. Prasanna, J. Reich, and D. Larner, "The abstract task graph: A methodology for architecture-independent programming of networked sensor systems," in *Proc. Workshop on End-to-End Sense-Andrespond Systems (EESR), in Conjunction with MobiSys*, Jun. 5, 2005, pp. 19-24.

[6] P. Levis, *et al.*, "TinyOS: An operating system for sensor networks," *AmbientIntell.*, pp. 115-148, 2005.

[7] R. Gravina, A. Guerrieri, G. Fortino, F. Bellifemine, R. Giannantonio, and M. Sgroi, "Development of body sensor network applications using spine," in *Proc. IEEE Int. Conf. Systems, Man Cybern.*, Oct. 2008, pp. 2810-2815.

[8] C. Lombriser, D. Roggen, M. Stäger, and G. Tröster, "Titan: A tiny task network for dynamically reconfigurable heterogeneous sensor networks," in *15. Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, ser. Informatikaktuell, Berlin, Germany: Springer, 2007, pp. 127-138.

[9] M. Zhang and A. Sawchuk, "A customizable framework of body area sensor network for rehabilitation," in *Proc. 2nd Int. Symp. Appl. Sci. Biomed. Commun. Technol. ISABEL 2009*, Nov. 24-27, 2009, pp. 1-6.

[10] V. Parthasarathy, P. Anandakumar, and V. Rajamani, "Design, simulation and FPGA implementation of a novel router for bulk flow TCP in optical IP networks," *IAENG International Journal of Computer Science*, vol. 38, no. 4, pp. 343-349, 2011.

[11] V. Parthasarathy and V. Rajamani, "Performance analysis of bulk flow TCP for routing in optical network," *Journal of Optical Communications*, vol. 29, no. 2, pp. 98-106, 2008.

Parthasarathy V is a professor in the Department of Computer Science and Engineering, Vel Tech Multi Tech, Chennai, India. He received his master degree in computer science and engineering from College of Engineering, Guindy, Chennai, Tamilnadu, India in 2004 and PhD from Anna University, Chennai, Tamilnadu, India in 2009. His research interest includes network, network security and sensor networks.

Sharmila P. is working as an assistant professor in the Department of Electronics and Communication Engineering, Vel Tech Multi Tech, Chennai, India. She received her M.E degree in applied electronics from C.Abdul Hakeem College of Engineering and Technology, Vellore, Tamilnadu, India in 2009. Currently she is pursuing her Ph.D. in wireless sensor Networks at Vel Tech University, Chennai. Her research interest includes sensor networks, mobile Ad hoc networks and signal processing.

Hemalatha S. received her master degree for Computer Science and Engineering from Anna University, Chennai, Tamilnadu, India. She is a research scholar pursuing her PhD in cryptography using ancient Indian Vedic Mathematics. She is working as assistant professor in the Department of Information Technology, Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai.