The need of Software Development Process for Wireless Sensor Networks with Cooperative Nodes

Gian Ricardo Berkenbrock
Federal University of Santa Catarina - UFSC
Software/Hardware Integration Lab.
Joinville, SC, Brazil
gian.rf@ufsc.br

Carla Diacui Medeiros Berkenbrock, Omir Correa Alves Junior
Santa Catarina State University - UDESC
Department of Computer Science
Joinville, SC, Brazil
{diacui,omir}@joinville.udesc.br

***Abstract*—Wireless sensor network (WSN) is a technology used to collect data from different sources. The development of WSNs offer complex challenges. Developers of WSN systems should be aware of issues such as energy-efficient network management and cooperation among nodes. The advancement in sensor nodes have created new opportunities for cooperation. This paper addresses the lack of software development process for WSN systems. In addition to that, we consider that cooperation among nodes could be successfully applied in wireless sensor networks. Nodes can cooperate to perform tasks such as verification of operation condition, improve the management of energy consumption and efficiency of data capture. However, we argue and propose cooperative considerations that should be investigated during the software development process.**

I. Introduction

Nowadays the goal of collecting data from different objects (for example, animals, people, cars and roads) is calling more attention. An example of project in this direction is uIPv6 [1]. This project is implemented as part of ContikiOS [2]. It enables embedded devices running ContikiOS to support IPv6. The usage of IPv6 facilitates the integration and interoperability with networks, information systems, and other embedded devices by considering the use of the same phy and mac layer protocol. Thus it addresses one requirement that is to support a larger address space in order to provide more flexibility in allocating addresses and routing traffic. In addition to that, IPv6 therefore is able to make the *Internet of Things* possible [3]. The Internet of Things is considered a complex system because it has to deal with features such as no predefined communication infrastructure, interactions between large variety of objects, changes over time, and capacity to identify instantaneously any kind of object.

Wireless Sensor Network (WSN) has been under research over at least two decades. The research interest has particularly increased in the last decade because of micro-electro-mechanical systems improvements that made feasible power tiny nodes to both sense and perform wireless communication.

WSN is composed of nodes that have capability to sense and to communicate over-the-air to each other. In addition, the nodes can perform local processing and have a local storage. In order to deliver the collected data to a base station in a multi-hop scenario, the nodes transmit the data using ad-hoc communication. Although with all these nodes characteristics, each one aggregate at least one constraint to the system such as short communication range, low bandwidth, small memory and limited energy. Then, typical scenarios for WSN application are environmental monitoring, surveillance (for example, border, home, building), object tracking, agriculture precision, bridge monitoring, hospital monitoring, herd (such as cattle, swine, horse) monitoring and so on.

In this paper we address the lack of software development process for WSN. We also explore some benefits and challenges of using cooperation for wireless sensor networks. The remainder of this paper is organized as follows. Section II describes issues related to software development process for dealing with WSNs requirements. Section III presents some ways to use cooperation in order to improve performance in WSN scenarios. Section IV describes requerimets to be considered during the software development process in order to aid project team increasing cooperation among nodes. Section V presents a sensor selection process to compose a task within the WSN. At last, in Section VI we present some final remarks and we discuss future work.

II. Software Development Process

Software development process [4] is a set of activities whose goal is software development or evolution. Thus, the usage of process to develop software assists the developer from software specification until deployment. As a generic view, there are three activities in such development process that is specification, development, evolution [4]. There are different lifecycles based on available process models, for example, waterfall process, incremental development, evolutionary development, iterative development, spiral model, prototyping, agile methods, and cots-based development. One can use any of them to develop ordinary software, i.e., software commonly used in desktops or servers. Even with some process adaptation or extension available, for instance, RUP extension for distributed systems [5], using MDA for embedded systems [6], and hybrid approach with waterfall and iterative development [7], there is a lack of process to address a development process for wireless sensor networks.

Besides the development of an application for running on nodes, the development process must also concern on a data model that describes all dataflow for in and out directions of WSN. This dataflow is one interest of process. The other interest is the simulation used by developers to evaluate their solutions and, in this case, simulation can play an important role through the design and implementation activities. One can use simulation in order to minimize type 3 [8] or type 0 error [9], then an ideal development process should help the developers to build systems as illustrated in Fig. 1.

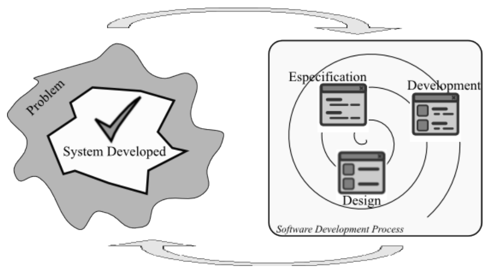

Figure 1. Software development process from problem to solution developed

When developing applications for WSN developers must also beware with characteristics such as application-oriented nature for software and hardware, limited battery, limited memory, coverage area, and time synchronization. The developers also need to take into account characteristics such as are hybrid, heterogeneous, base station quantity, energy efficient network management, and cooperation among the nodes to perform their tasks.

Cooperation provides a more convenient interaction among members of a group [10]. It is possible to use cooperative systems to help developers to work together during the software development process. In this way, cooperative systems can be designed to accommodate practices related to the software development process. We believe that productivity can be improved through collaboration among developers and we intend to analyze this issue in further work. However, cooperation can take many forms. Another alternative is to think of development of WSN systems with the cooperation of nodes. In this alternative, for instance, it is possible to explore different cooperation strategies among nodes in order to obtain capacity improvement in WSNs.

III. Cooperative Issues

The advancement of sensor nodes have created new opportunities for cooperation. However, due to the restriction of both sensor nodes and wireless network enabling cooperation among the devices, in an extremely dynamic context, poses real challenges. In particular, we have to deal with resource constraints such as energy, processing power, memory, high failure rate and node mobility. Furthermore, in [11] it is mentioned that the signal-to-noise ratio (SNR) also has a significant impact on the cooperation.

Moreover, sensor nodes may have either single or multi-hop communication, may be awake or sleep, and may use mobile or fixed devices. This wide variance makes difficult to define which features should be considered in order to provide cooperation among nodes. The set of features should be analyzed and implemented during the software development process.

Cooperation requires communication and coordination. Effective communication is vital to successful WSN, however, the finite energy source in WSNs makes hard to provide it. Due to the fact that WSNs are composed of a large number of energy constrained sensor nodes, and the nodes can enter in sleep state in order to save energy, the communication can be disruption. The effectiveness of communication and cooperation can be enhanced if nodes tasks are coordinated. The coordination is the integration of nodes efforts towards the accomplishment of a goal. Without coordination sensor nodes can engage in conflicting or repetitive actions.

Although all the challenges in supporting cooperation among nodes, cooperation would be successfully applied in wireless sensor networks. Cooperation can help to determine whether a deployed network is functioning correctly. For instance, Tole and Culler [12] present a cooperative management system for WSNs that provides a query system to enable rapid acquisition of network health and performance data, and a logging system to enable recording and retrieval of events generated by system. Maric and Yates [13] propose a cooperative strategy in order to increase the energy-efficiency and to facilitate load balancing.

Couderc [14] investigates the collaborative capture. According to Couderc, collaborative capture reveals new perspectives in ambient computing. Sensor can capture sound, picture, video, can cooperate with other sensors such as GPS positioning receivers and it can communicate with global as well as local networks. Collaborative capture can also be used to increase the quality and the accuracy of a given data, by combining the data of several nodes.

In [15] it is presented a model of cooperative work designed for WSNs. In this model, collaboration among nodes can be based on a set of properties such as proximity, common cluster, phenomenon to monitor and hardware characteristics of the sensor node. Regarding collaboration, the model includes some Computer Supported Cooperative Work (CSCW) concepts and presents an awareness tool in order to allow an interactive navigation in a network map.

Banitalebi et al. [16] propose an algorithm to optimize the collaborative data communication algorithms in terms of energy efficiency, reliability and life time of sensor nodes. This algorithm focus on minimizing the energy consumption by decreasing the number of collaborative nodes and at the same time guarantees the demanded quality by selecting the higher quality nodes to participate in the collaboration.

In [17] it is presented a Cooperative Network Coding (CNC) proposed to optimize and balance the use of forward error control, error detection, and retransmissions at the packet level for these networks. CNC combines cluster-based cooperative communication and network coding with the aim of improving the communication reliability.

WSN may consist of hundreds or thousands of nodes. Cooperative groups for WSN can vary significantly in terms of number of nodes. Interaction between many nodes engaged in tasks tends to cause bandwidth congestion as well as inefficient use of node resources such as energy. Then, an efficient data dissemination technique should work well not only in small-scale sensor networks but also in large-scale ones. In order to avoid unsuccessful cooperation among nodes, it is important to specify a data dissemination scheme that can provide a good balance among reliability, scalability and energy efficiency.

WSNs are, by nature, collaborative networks. However, as we have seen, cooperation in WSN pose many technical issues. As we have also discussed the software development process are not enough to design and implement successful WSNs.

IV. Cooperative Requirements for Software Development Process

Software development process can be seen as a set of guidelines that assist a team to perform the work in a disciplined way from the concept to the development of a give solution. In what follows we suggest a set of requerimets to be considered during the software development process in order to aid project team to increase the cooperation among nodes.

The success of collaboration depends on aspects like communication, coordination and context. Based on these three aspects, we formulated categories of cooperative requirements for software development process.

Communication has an important role in collaborative activities. For exemple, if a sensor loses track of its possible partner, collaborating with them becomes much more difficult. Communication is also a way in which context information and other messages can be exchanged among nodes.

A number of communication properties can be used during the software development process: interaction pattern (one-to-many, many-to-many, many to one, clustering), state (active, inactive, stand-by), coverage (area, barriers crossing path), transmission cost (i.e., power consumption, latency), delay, reliability (i.e., accuracy, predictable levels of performance), integrating with wake/sleep schedules, scope (unicast, multicast and anycast semantics), mobility (static, mobile nodes, mobile sinks), addressing (physical, logical) and congestion.

Coordinating actions in a collaborative activity means making actions happen in the right order, at the right time, and generally, making them meet the constraints of the task. Depending upon the cooperative goal, coordination can be necessary at several levels of granularity, from small movements to large-scale divisions. We list some important coordination properties to allow for more effectiveness and usability for: QoS requeriments (requirements specified by the application such as optimal formation of clusters, total energy consumption, energy load balance in the system and time/energy cost model), resource allocation (manages network resources such as remaining energy of sensors, network connectivity, node capability, and cluster formation from distributed sensors), time synchronization (periodic, event-triggered), number of collaborative nodes, human interface (for instance, what should be shown in the control station screen), goal (sense-only, sense-and-react) and application requeriments (for instance, collaboration between sensor nodes and other devices or features to represent a certain scenario).

Finally, context is important to help sensors to recognize opportunities for cooperation. It can be used in order to reduce the effort needed for communication, as well as to simplify the coordination, by allowing sensor to act in anticipation of others, and by providing information about important events for appropriate help and assistance. Context is also useful in the coordination and division of activities. As the task progresses, sensor can regularly reorganize what each sensor will do next. Then, the activity can be planned and replanned in a better way. We suggest a set context properties for improving cooperation in the development of WSN systems: resource context (memory, processing capabilities, sensing capabilities, hardware), energy context (battery), localization, space (the processing involves the whole network or it occurs only within some limited area) and awareness (explicit, implicit).

The cooperative requirements presented in this section can be used to define a Collaboration Opportunity (CO). This CO is used to help the selection of appropriate sensors to compose a task within the WSN. Further details about the proposed sensor selection process are shown in the next section.

V. Requirements for Sensors selection

The process of analysing and selecting the most appropriate sensors to compose the WSNs is a critical phase in the WSNs life cycle, because the success of the operation depends on how the sensors are selected [12].

In this section we propose a method for selecting the most appropriate sensors to compose a WSN. This method is supported by a pre-qualified sensor database and the Analytic Hierarchy Process (AHP) [18]. The method itself starts after the Collaboration Opportunity (CO) arrival. A CO, besides other information, is composed of one or more tasks and each task is supported by a group of sensors.

The Fig. 2 depicts an example for selecting the sensors. In this example the CO is composed of 3 tasks and each task can be deployed by different groups of sensors, and one group per task will be chosen to deploy the WSN. The sensor selection process is shown in phases.

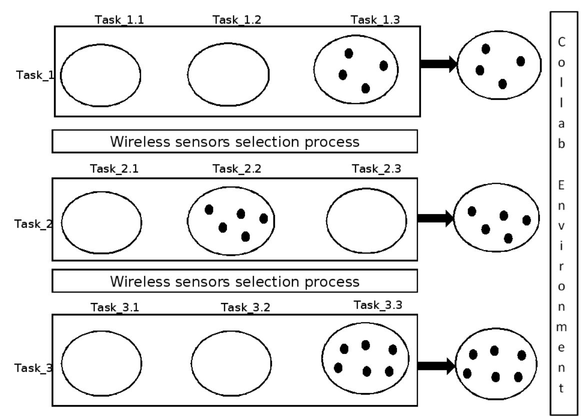

Figure 2. Sensors selection process

Phase 01: In this phase the number of tasks per CO area are identified and for each task the method selects one or more sensors. The sensors are autonomous members of this database. Since they share principles and working methods, the creation of WSNs from a pre-qualified sensor database becomes faster, more effective and less complex to manage. A task to be performed by sensor is composed as a list of competencies, the sensors are identified based on these competencies and allocated in a queue as depicted in Fig. 3. This process is repeated for each task.

Phase 02: For each sensor pre-selected and allocated in the queue the proposed method calculates the number of presence

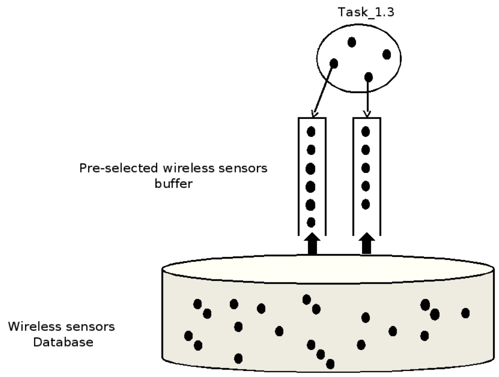

Figure 3. Sensors database and selection queue

Key Performance Indicator (KPI). The number of presence is a strategic metric that indicates the sensor level of commitment. At each pre-defined period of time the method checks if the sensor is operational (1) or not (0) and update the table as shown in Fig. 4. So, the samples are summarized (for example S1, S2 and S3) as vectors of samples.

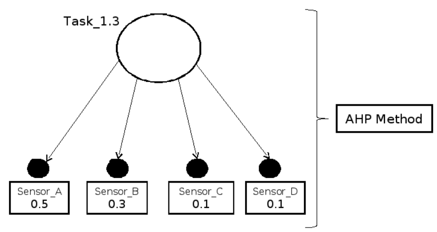

Figure 4. Number and vector of samples matrix

Phase 03: The AHP method [18] is applied in the vector of sample matrix as depicted in Fig. 5. Each sensor has a level of importance in the WSN and the main purpose of the AHP method is to allow assigning different weights to sensors according to their characteristics and level of importance.

	Number of sensors				
AHP matrix	1*0.4	3*0.3	3*0.1	5*0.1	2*0,1
	0*0.4	4*0.3	5*0.1	4*0.1	4*0.1
	2*0.4	3*0.3	3*0.1	4*0.1	5*0.1

Figure 5. AHP method applied to sensors

Phase 04: The sensors are measured based on the sum of each columns AHP matrix, as depicted in Fig. 6, and the highest values are selected to compose each task of the WSN (Fig. 2). For example in Fig. 6, for sensors 1, 2, 3, 4 and 5, was assigned the weights 0.4, 0.3, 0.1, 0.1, 0.1 respectively.

VI. Conclusions and Future Work

This paper presents a discussion about the lack of suitable process to address a development process for wireless sensor networks. Furthermore, we argue that it is possible to explore cooperation among nodes in order to obtain capacity improvement in WSNs. In this way, cooperation could be considered during the software development process.

		Number of sensors				
Number of samples	1	0	0	1	1	1
	2	0	1	1	1	0
	3	0	0	0	1	0
	4	0	1	0	1	0
	5	1	0	0	1	0
	6	0	1	1	0	1
	Σ	1	3	3	5	2
Vectors of samples	S1	1	3	3	5	2
	S2	0	4	5	4	4
	S3	2	3	3	4	5

Figure 6. AHP matrix

References

[1] Durvy, M., Abeill, J., Wetterwald, P., O'Flynn, C., Leverett, B., Gnoske, E., Vidales, M., Mulligan, G., Tsiftes, N., Finne, N. , Dunkels, A.: Making Sensor Networks IPv6 Ready. In: Proceedings of the Sixth ACM Conference on Networked Embedded Sensor Systems, 2008.

[2] Dunkels, A., Grönvall, B., Voigt, T.: Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In: Proceedings of the First IEEE Workshop on Embedded Networked Sensors, 2004.

[3] Dunkels, A., Vasseur, J.: IP for Smart Objects. In: Internet Protocol for Smart Objects (IPSO) Alliance, 2008.

[4] Sommerville, I.: Software Engineering, 7th ed, Addison-Wesley, 2004

[5] Paes, C. E. B., Hirata, C. M., Yano, E. T.: Extending RUP to develop fault tolerant software. SAC 2008, pp .783–790

[6] DeAntoni, J., Babau, J.P.: A MDA-based approach for real time embedded systems simulation. In: Proceedings of IEEE International Symposium on Distributed Simulation and Real-Time Applications, 2005.

[7] A Software Development Process for Small-Scale Embedded Systems, Embedded technology magazine. Feb, 2008.

[8] Balci, O., Nance, R.E.: Formulated problem verification as an explicit requirement of model credibility. Simulation 45(2), 1985, pp 76–86.

[9] Pidd, M.; Computer Simulation in Management Science. 5 ed., John Wiley & Sons, 2004

[10] Berkenbrock, C. D. M. and Hirata, C. M.: Supporting Cache Coherence in Mobile Cooperative Systems. In: The 7th IEEE International Symposium on Network Computing and Applications, 2008.

[11] Ng, C.T.K., Goldsmith, A.J.: Capacity and Cooperation in Wireless Networks. In: Proceedings of Information Theory and Applications, 2006.

[12] Tolle, G., Culler, D.: Design of an application-cooperative management system for wireless sensor networks. In: Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.

[13] Maric, I., Yates, R. D.: Cooperative Multicast for Maximum Network Lifetime. In: IEEE Journal on Selected Areas in Communications, 2005.

[14] Couderc, P.: Collaborative Capture: A New Perspective for Sensor Networks. In: European Research Consortium for Informatics and Mathematics Number 67, October 2006.

[15] Peralta, L.M.R. and Brito, L.M. and Reis, M.D.L.: Towards a Collaborative Model for Wireless Sensor Networks. In: IFIP International Federation for Information Processing, vol. 283, no. 1, 2010.

[16] Banitalebi, Behnam and Miyaki, Takashi and Schmidtke, Hedda R. and Beigl, Michael.: Self-optimized collaborative data communication in wireless sensor networks. In: Proceedings of the 2011 workshop on Organic computing, 2011, ACM.

[17] Arrobo, G. E.; Gitlin, R. D. Minimizing Energy Consumption for Cooperative Network and Diversity Coded Sensor Networks. In: 4th IEEE Latin-American Conference on Communications 2012. Cuenca, Ecuador, 2012

[18] Saaty, T. L. How to make a decision: the analytic hierarchy process. European Journal of Operational Research, v. 38, n. 1, p. 9-26, 1990.