Distributed Intelligent Rule-Based Wireless Sensor Network Architecture

A. Cubero-Fernandez, J.M. Castillo-Secilla, Jose M. Palomares, J. Olivares,
and F. Leon

Author version of the paper published in Springer:
https://link.springer.com/chapter/10.1007/978-3-319-00566-9_24
Cite as: Cubero A., Castillo J.M., Palomares J.M., Olivares J., León F. (2013) Distributed Intelligent Rule-Based Wireless Sensor Network Architecture. Ambient Intelligence - Software and Applications. Advances in Intelligent Systems and Computing, vol 219. Springer, Heidelberg. https://doi.org/10.1007/978-3-319-00566-9_24

Abstract This paper describes the design of a new system architecture for monitoring and controlling purposes of a group of sensors and actuators within a wireless sensor network (WSN). This system can manage an undefined amount of clustered networks. The proposed system architecture enables Internet communications to reach the WSN in a highly efficient way. This structure reduces the bottleneck of the Internet/WSN bridge and the amount of messages inside the WSN when an Internet request arrives. Besides, each individual WSN implements an Intelligent Rule-Based System Automation (IRBSA) that performs the automation of the behaviour of the network motes according to the previously included rules. These rules describe the actions that are executed when all the conditions of that rule are met. Opposite to traditional approaches, IRBSA is placed in the WSN Header Mote rather than in the Internet server or in every mote.

1 Introduction

Nowadays, wireless comunications are continuously growing both in industry and in domestic environments. It is easy to make an effective and secure wireless sensor network (WSN) compared with wired networks. Currently, many WSN are opening their capabilities to the Internet, offering the data acquired by the sensors included within the WSN to the World Wide Web. Many efforts have been made by researchers in this field, paying special interest in the conception of the Internet of Things (IoT) [7]. This approach tries to interconnect any wired or wireless device, even with different types of hardware and physical network modems. The first

---

Antonio Cubero Fernández · José María Castillo Secilla · José Manuel Palomares Muñoz ·
Joaquín Olivares Bueno· Fernando León García
Computer Architecture, Electronics, and Electronic Technology
University of Cordoba, Cordoba, Spain
e-mail: acuberof@gmail.com, {jmcastillo,jmpalomares,olivares}@uco.es, fndoleon@gmail.com

step to get an Internet–conected WSN is to have a front–end server (usually a Web server) acting as a bridge between the Internet and the WSN. This structure is very suitable for any WSN with a small number of nodes (usually, called motes) and with very few requests from the Internet side. Each time an Internet request arrives at the Web server, the computer translates the Internet query to the WSN, with a message to a specific device address, requesting the value of an input pin, or assigning a value to an output pin, etc. However, this structure collapses when the WSN enlarges or when the amount of Internet requests rise.

Next stage in the WSN development roadmap is to build fully distributed systems, in which system workload is shared among multiple devices. Besides, fully distributed WSNs allow to introduce fault–tolerance. If any component fails, it does not affect the whole system. Some mechanisms are included in the WSN infraestructure to change the role of an idle mote to handle the tasks of the faulty one. Some other benefits may be obtained from fully distributed WSN, for example, these systems allow to enlarge dinamically the amount of motes in an easy way. Thus, this makes the systems to be stronger and more powerful.

The main objective of this project is to design a distributed and intelligent system to manage multiple WSN. Each WSN acts independently, although, some of them are joined to work cooperatively forming larger WSN. Each WSN has its own Intelligent Rule–Based system being in charge of automation of the motes of that very WSN. A communication platform will be created for the interconnection of the WSN. Each WSN has a WSN Header Mote (WHM), which acts as a Coordinator device of that WSN. Each WHM is able to communicate directly with the communication platform. This communication platform makes possible to handle the data extracted from every WSN. However, the communication platform also offers a entrance gate for the Internet to the whole system, to request or update any data of any mote in any of the WSNs. The communication platform is optimized using databases to store the data about WSNs. This fact allows much faster readings of the data of the sensors linked to a mote, without the necessity to ask directly the involved mote to obtain the data. This view of the database will be replicated in each WHM to make every WSN completely self–sufficient.

This article has been divided in several sections. Section 2 makes a short scientific revision about current articles related to the proposed system. The proposed System Architecture is described in Section 3. The Intelligent Rule–Based System Automation is stated in Section 4. Some prototypes are described in Section 5. The conclusions of this work and some future work are presented in Section 6.

2 Scientific Review

Many researchers have used WSN to monitor and control enviroments. Some authors have designed very interesting systems in order to communicate WSN with the Internet while collecting large data from sensors. André *et al.* [3] proposed a model for monitoring WSNs based on a REST Web service and XML messages to provide a mobile ubiquitous approach for WSN monitoring. Data collected from WSN are stored in a database, although every request from the Internet produces a WSN message to the end–node which has the requested sensor.

Serdaroglu and Baydere [6] studied a proxy–based and gateway–based system. They defined an hybrid approach which combines the advantages to interconnect WSN and the IP networks. The proposed approach is used to build a web server for WSNs. The goal of the study is to reduce memory footprint of the overall system and use possibly small amount of resources of a WSN node implementing in a middleware rather than a full conventional stack or a ready solution.

Previous approaches make the data analysis in devices that are not inside the WSN, either in remote nodes in the Internet or in the Internet proxy or gateway. For these models, WSN are non–intelligent and any Ambient Intelligence strategy must be addressed by elements outside the WSN.

The opposite approach is to include an intelligent data processing in every mote, making the WSN not be a mere network infraestructure, but an intelligent one. Labraoui *et al.* [5] proposed a new scheme for data aggregation in large–scale WSN. However, authors focused mainly in fault–tolerance in clustered–based WSN.

Tapia *et al.* [8] proposed an innovative platform that addressed the requirements of Ambient Intelligence paradigm, such as context–awareness and ubiquitous communication, allowing the use of heterogeneous WSNs and taking advantage of the use of intelligent agents directly embedded on wireless nodes. This article described the integration of the HERA (Hardware-Embedded Reactive Agents) platform into FUSION@ (Flexible and User Services Oriented Multi–agent Architecture) [1, 9]. This way, through the integration of HERA and FUSION@, there was no difference between a software and a hardware agent.

Another issue to be analyzed are the Rule–Based Systems in WSN. Rule–Based system are used as a way to store and manipulate knowledge to interpret information in a useful way. They are often used in artificial intelligence applications and research. They consists of a rule–base (permanent data), an inference engine (process), and a workspace or working memory (temporary data). Knowledge is stored as rules in the rule–base (also known as the knowledge base).

Dressler *et al.* [2] designed a Rule–Based Sensor Network (RSN) that mimics the cellular signaling communication. That model has data–centric communications and the rule–based programming scheme describes specific actions after the reception of specific data fragments for simple local behaviour control. RSN is able to process sensor data and to perform network–centric actuation according to a given set of rules. In particular, this system is able to perform collaborative sensing and processing in SANETs with purely local rule–based programs.

All these systems are suitable for the required objectives. However, they demand computationally powerful platforms to be executed, and this work tries to minimize the computational requirements of the WSN nodes, while providing similar results in terms of performance.

3 System Architecture

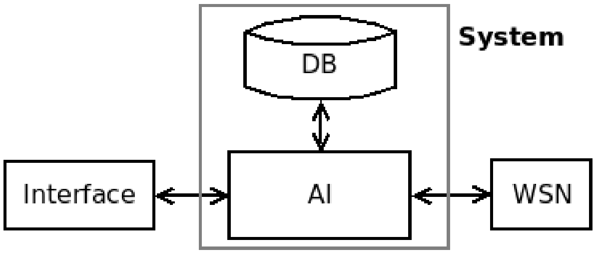

Fig. 1 Typical Internet–connected WSN system architecture

In this section, the system architecture is described. Traditional WSN (Fig.1) are compounded by an interface, an internal system application, and an access point to the WSN. The interface allows the user to input the requests and to receive the results. Internal system is usually composed by an application that performs the automation and controls all devices and a warehouse storing data. Finally, the WSN is formed by some motes, responsible for gathering data from the environment. This structure is not very scalable and it can be easily overloaded. Therefore, every request from the interface (external orders) would be translated into, at least, one internal message in the WSN, even though of the requested value has not changed. Thus, large amount of requests arriving from the interface would degrade the wireless medium. It is obvious that this structure lacks generality and a different solution is requiered. For that reason, a fully distributed system, called DIRB–WSN (Distributed Intelligent Rule-Based – Wireless Sensor Network), is proposed (Fig. 2). To achieve this objective, the system is divided into two different modules: *Communication_AI* and *WSN_AI*.

Communication_AI manages input connections from the Internet and the subsequent responses. This module reduces the amount of messages to WSN using

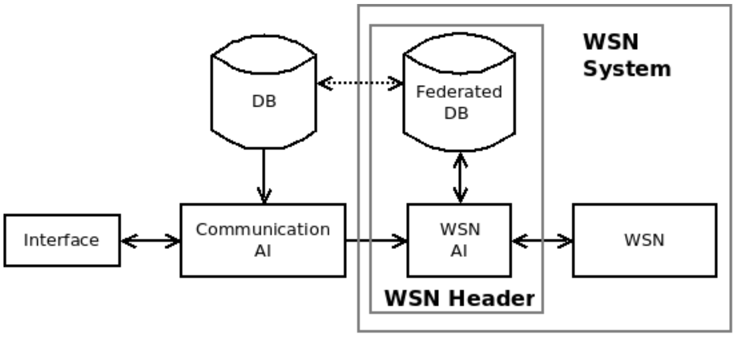

Fig. 2 Proposed Internet–conected DIRB–WSN system architecture

federated tables for each WSN. The Database Managemente System (DBMS) is in charge of synchronizing the data between database copies allocated in different devices. When a reading sensor query is requested, *Communication_AI* looks for the data in its own copy of the federated table, so that, no message is sent to the WSN. This simple, but very effective, mechanism simplifies the working process of the *Communication_AI*. On the other hand, if a writing or update action is requested by the user, *Communication_AI* sends a message to the *WSN_AI* requesting the modification of the data for a given mote. *WSN_AI* modifies its copy of the federated table (therefore, the *Communication_AI* federated table is also modified by the DBMS) and a message is sent to the wireless mote to modify or update an output value. *Communication_AI* also stores a log for each WSN.

WSN_AI is placed in the Header Mote of every WSN (WHM). This mote is usually the coordinator node of the WSN and it also has another network interface which is able to connect to the *Communication_AI* node. *WSN_AI* is in charge of analyzing and processing the input packets coming from *Communication_AI* and from the inner WSN. It also runs IRBSA (Section 4).

4 Intelligent Rule-based System Automation

The second part of the project consists of a soft real–time fully automation of the system. For the sake of simplicity, the WSN is assumed to have some kind of time–synchronization protocol, therefore, the time/date of all the motes in the WSN is coherent. The Intelligent Rule–based System Automation (IRBSA) allows to program different behaviours by checking some parameters of the system. IRBSA is composed of a set of rules to be executed locally within each WSN.

Every IRBSA rule have two different parts: *antecedents* and *consequents*. An *antecedent* holds the conditions that must be true for the rule to be executed. Each *antecedent* is composed of the following elements:

- *Active*: it indicates whether an antecedent has to be examined.
- *Date*: date after which the antecedent is evaluated.

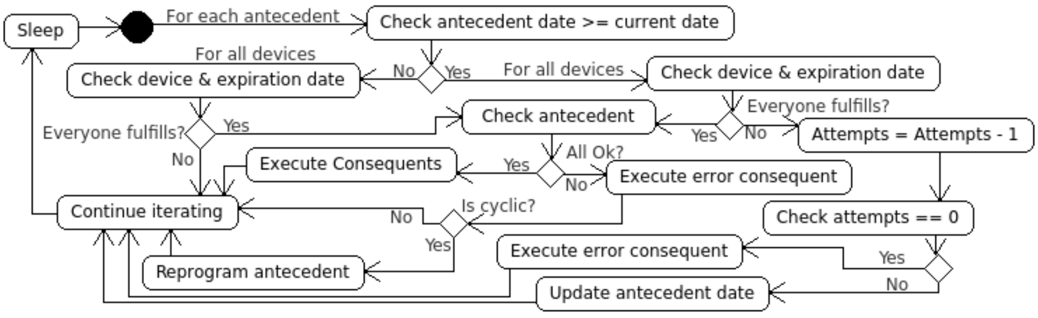

Fig. 3 IRBSA Activity Diagram

- *Period*: time interval for which the antecedent is enabled.
- *Attempts*: maximum number of attempts that the antecedent will be analized. An attempt expires when *period* ends and one or more devices are not still ready.
- *Cycle*: it indicates whether an antecedent date has to be rescheduled with a new future date.
- *Rule*: it states a dependency between the antecedent and one or more rules.
  - These rules must have been activated before the antecedent is checked.
- *Device*: it states a dependency between the antecedent and one or more devices (or other physical constraints).
  - This element imposes a condition that a device must fulfill to activate the rule. For example, the pin value of a device must be greater than another value.

On the other hand, *consequent* indicates an action to be executed in the system. This action ranges from a simple change of the state of an actuator to the reprogramming of a mote, and even to modify the IRBSA. There are two different *consequent* types: *normal consequent* and *error consequent*. Once IRBSA checks an *antecedent*, every rule and device dependencies are analyzed. If an *antecedent* of a given rule is not provided, the *error consequents* associated to the rule are executed. Whereas, if all the *antecedents* are met, *normal consequents* are executed. Besides, if any of the *antecendents* of the rule are cyclical, new time instances of all the *antecedents* are rescheduled. This fact assures that recent values are used for the *antecedents* before the *deadline* of the rule. Figure 3 shows the activity diagram of IRBSA.

5 Results

The proposed DIRB–WSN system is a complete functional system, composed of a Internet server with the *Communication_AI* and some WSN systems headered by a *WHM* managing a network of motes. A prototype has been deployed using XBee modules (Zigbee networking) placed on Arduino boards to create the WSN. Figure 4 shows a mote prototype, with a humidity/temperature sensor, a movement sensor, some buttons and switches, and a servo–motor connected to an Arduino FIO board with an XBee modem.

This model has not been compared to any other previous proposals, due to time restrictions. However, many tests have been carried out with different workload and Internet requests. All the tests have provided similar results in terms of number of messages within the WSN. A test scenario was deployed: the WSN was composed of 3 XBee devices, two mote prototypes and a WHM. The WHM was a notebook computer with an Xbee modem and a WiFi port running the *WSN_AI* and a copy of the *federated database*. This WHM was interconnected to a Web server computer. This Web server was running also the *Communication_AI* and the main *federated database*. The experimental test was to simulate 100 Internet queries (50 of them, sensor reading requests and 50 of them sensor updating messages) to the Web server for 20 seconds. One of the mote prototypes obtained the temperature from the sensor once every second. Therefore, the number of messages in the WSN was 70: 20 of them starting from the prototype mote due to the sampling of the temperature sensor and 50 of the messages were because of the modification requests from the Internet side. As the number of messages was relatively low, no collapsing of the WSN was provoked and there were no loss of messages. On the other side, using a standard structure (as described in Fig. 1) would provided a minimum amount of 100 WSN messages (one WSN packet for each Internet request).

Regarding IRBSA, results show that it is a highly efficient system with a quick response to any stimulus, taking less than one second to provide a response in the worst case. This module is able to reduce the amount of messages going outside the WSN to the Internet, where the Ambient Intelligence module could be placed. A simple experimental test was evaluated, by including a rule in the IRBSA in which several sensor values of the prototype were taken into account to activate the servo–motor. For a 60 seconds experiment, the proposed system was able to reduce the number of Internet packets to zero. Meanwhile, for the same experiment, the standard structure with the Intelligent module as a remote application in the Internet side, produced more than 60 Internet messages (more than one each second). Besides, due to WSN collisions and Web server delays, the execution of rules in the standard structure experiment were delayed some seconds, most times.

6 Conclusions and future work

This system allows to manage multiple WSN Systems efficiently. The proposed infraestructure and the IRBSA make possible to automate these WSN Systems, which can be managed from a single Internet–conected machine. IRBSA is able to provide some kind of intelligence, keeping very low the complexity of the end–motes with-

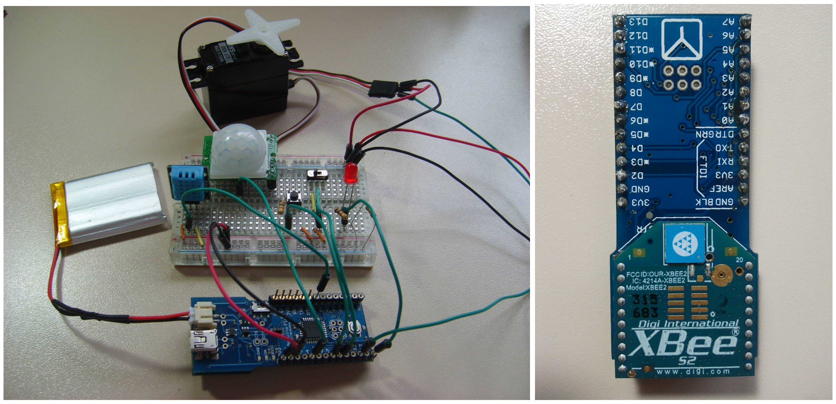

Fig. 4 Mote prototype

out provoking a bottleneck in the Internet gateway connection.

The usage of *federated tables* in distributed databases is a smart mechanism to disseminate information very efficiently. The DBMS is responsible for the synchronization, thus, the code of the *Communication_AI* is simplified as there is no need of checking for concurrent accesses. Furthermore, the data synchronization messages between databases may rely on a different data network, and thus, the amount of messages inside the WSN for reading requests from the Internet can be reduced drastically.

Related to the future work, it is very interesting to include fuzzy logic control (FLC) into IRBSA, in order to evaluate the rules taking both approximated *antecedents* and *consequents* rather than fixed and exact ones. FLC provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information [4].

It is planned to make more experiments and to compare the proposed DIRB–WSN with previous proposals. The experiments will be real scenarios with different deployed WSN.

References

1. Corchado, J., Bajo, J., Tapia, D., Abraham, A.: Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare. IEEE Transactions on Information Technology in Biomedicine **14**(2), 234 –240 (2010). DOI 10.1109/TITB.2009.2034369
2. Dressler, F., Dietrich, I., German, R., Krüger, B.: A rule-based system for programming self-organized sensor and actor networks. Computer Networks **53**(10), 1737 – 1750 (2009). DOI 10.1016/j.comnet.2008.09.007
3. Elias, A., Rodrigues, J., Oliveira, L., Zarpelão, B.: A ubiquitous model for wireless sensor networks monitoring. In: Proceedings - 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012, pp. 835–839 (2012)
4. Feng, G.: A survey on analysis and design of model-based fuzzy control systems. IEEE Transactions on Fuzzy Systems **14**(5), 676 –697 (2006). DOI 10.1109/TFUZZ.2006.883415
5. Labraoui, N., Gueroui, M., Aliouat, M., Petit, J.: Rahim: Robust adaptive approach based on hierarchical monitoring providing trust aggregation for wireless sensor networks. Journal of Universal Computer Science **17**(11), 1550–1571 (2011). Http://www.jucs.org/jucs_17_11/rahim_robust_adaptive_approach
6. Serdaroglu, K., Baydere, S.: Seamless interconnection of wsn and internet. In: 2012 20th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2012, pp. 1–6 (2012)
7. Tan, L., Wang, N.: Future internet: The internet of things. In: ICACTE 2010 - Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering, vol. 5, pp. V5376–V5380 (2010)
8. Tapia, D.I., Fraile, J.A., Rodríguez, S., Alonso, R.S., Corchado, J.M.: Integrating hardware agents into an enhanced multi-agent architecture for ambient intelligence systems. Information Sciences **222**(0), 47 – 65 (2013). DOI 10.1016/j.ins.2011.05.002
9. Tapia, D.I., Rodríguez, S., Corchado, J.M.: A distributed ambient intelligence based multi-agent system for alzheimer health care. In: A.E. Hassanien, J.H. Abawajy, A. Abraham, H. Hagras (eds.) Pervasive Computing, Computer Communications and Networks, pp. 181–199. Springer London (2010). DOI 10.1007/978-1-84882-599-4_9