Hindawi Publishing Corporation
Journal of Sensors
Volume 2014, Article ID 782789, 13 pages
http://dx.doi.org/10.1155/2014/782789

Review Article

Autonomic Wireless Sensor Networks: A Systematic Literature Review

Jesús M. T. Portocarrero,[1] Flávia C. Delicato,[1] Paulo F. Pires,[1] Nadia Gámez,[2] Lidia Fuentes,[2] David Ludovino,[3] and Paulo Ferreira[3]

[1] *PPGI-iNCE/DCC-IM/Federal University of Rio de Janeiro, Rio de Janeiro, Brazil*
[2] *Departamento de Lenguajes y Ciencias de la Computación, University of Malaga, Málaga, Spain*
[3] *INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal*

Correspondence should be addressed to Jesús M. T. Portocarrero; jesus140@gmail.com

Received 16 October 2014; Revised 29 November 2014; Accepted 29 November 2014; Published 29 December 2014

Academic Editor: Andrea Cusano

Autonomic computing (AC) is a promising approach to meet basic requirements in the design of wireless sensor networks (WSNs), and its principles can be applied to efficiently manage nodes operation and optimize network resources. Middleware for WSNs supports the implementation and basic operation of such networks. In this systematic literature review (SLR) we aim to provide an overview of existing WSN middleware systems that address autonomic properties. The main goal is to identify which development approaches of AC are used for designing WSN middleware system, which allow the self-management of WSN. Another goal is finding out which interactions and behavior can be automated in WSN components. We drew the following main conclusions from the SLR results: (i) the selected studies address WSN concerns according to the self-* properties of AC, namely, self-configuration, self-healing, self-optimization, and self-protection; (ii) the selected studies use different approaches for managing the dynamic behavior of middleware systems for WSN, such as policy-based reasoning, context-based reasoning, feedback control loops, mobile agents, model transformations, and code generation. Finally, we identified a lack of comprehensive system architecture designs that support the autonomy of sensor networking.

1. Introduction

Wireless sensor networks (WSNs) consist of networks composed of devices equipped with sensing, processing, storage, and wireless communication capabilities. Each node of the network can have several sensing units, which are able to perform measurements of physical variables, such as temperature, luminosity, humidity, and vibration [1]. The nodes in a WSN have limited computing resources and are usually powered by batteries; thus energy saving is a key issue in these networks in order to prolong their operational lifetime. WSN nodes operate collaboratively, extracting environmental data, performing the same simple processing, and transmitting them to one or more exit points of the network (often called sink nodes), to be analyzed and further processed.

There is currently a wide range of applications for WSN, ranging from environmental monitoring to structural damage detection. The quality of a WSN application depends not only on how well it has been designed and implemented but also on how well it can deal with problems and events at runtime [2]. Typically, WSNs are used in highly dynamic and sometimes remote and/or hostile environments and should operate without or with minimal human intervention. Therefore, such networks should have an autonomous behavior and be able to tolerate several types of failures, such as faulty nodes or hardware physical malfunction (e.g., failures in the sensor units or battery) and lack of coverage and connectivity, among others. In other words, WSN should be able to self-manage those failures and to dynamically self-adapt to the environment [3].

The first WSN applications had simple requirements that did not demand complex software infrastructures. Typically, WSNs were designed to meet the needs of a single target application usually of a single user, who was also the infrastructure owner. However, with the rapid evolution in this field combined with the increasing complexity of sensors and applications, the need of specific middleware platforms for these networks has risen [4]. A WSN middleware is layered software that lies between application code and the communication infrastructure providing, via well-defined interfaces, a set of services that may be configured to facilitate the application development and its execution in an efficient way for a distributed environment [5]. Thus, the main goal of a middleware is to enable the interaction and the communication between distributed components, hiding from application developers the complexity of the underlying hardware and network platforms, and freeing them from explicit manipulation of protocols and infrastructure services. WSN middleware should provide generic services for applications based on sensing and additionally consider application-specific needs and the inherent features of WSN nodes, such as the nodes limited resources of energy, memory, and CPU and the dynamic execution context. Middleware systems developed until today (e.g., [6–10]) represent good instruments for defining the high-level application logic and to deal with heterogeneity and distribution issues, but most of them do not provide an explicit way for defining the underlying autonomic behavior.

In order to grant autonomic behavior, individual components of any autonomic system should foresee the following set of functionalities, also known as self-* properties [36]: self-configuration, self-healing, self-optimization, and self-protection. Self-configuration is the ability of a system to adapt itself to the environment, changing according to high-level policies, aligned with business goals and defined by system administrators. Self-healing is the ability of a system to recover after a disturbance and to minimize interruptions to maintain the software available for the user, even in the presence of individual failure of components. Self-optimization is the system's ability to improve its operation continuously. And self-protection is the ability to predict, detect, recognize, and protect from malicious attacks and unplanned cascade failures.

These properties are the essence of autonomic computing. According to [36], autonomic computing (AC) is the capacity of an infrastructure for adapting itself according to policies and business goals. The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement. A highlighted approach to develop autonomic systems is the architecture for AC proposed by IBM [37] that defines an abstract framework for self-managing IT systems. In this framework, an autonomic system is a collection of autonomic elements. Each element consists of an autonomic manager and a managed resource. The autonomic manager allows adaptation through four activities: monitoring, analyzing, planning, and executing, with support from a knowledge base. In the monitoring activity, elements collect relevant data via sensors to reflect the current state of the system, the managed resource (and thus providing it with context awareness). In the analyzing activity, the current state of the context and of the managed artifacts is evaluated and if undesired states are detected a new (desired) target state is specified. The planning activity decides the necessary steps to adapt the system to move it from the current state to the desired state. In the execution activity the adaptation actions determined by the planning activity are executed by actuators or effectors. In order to achieve self-adaptation, software feedback loops are required, with explicit functional elements and interactions between them for managing the dynamic adaptation. These elements are known as MAPE-K model (monitor, analyze, plan, execute, and knowledge base) [37].

Considering its characteristics, AC is a promising option to meet basic requirements in the WSN design. Autonomic computing principles can be applied to a WSN in order to optimize network resources, facilitate its operations, and achieve the desired functionality in the wide field of sensing-based applications besides providing conditions for this type of network to manage itself without requiring human operators. The application of these AC principles into WSN would be facilitated by the development of a system at the middleware level. Therefore, we believe that a study addressing a comprehensive analysis about middleware proposals for autonomic WSN is quite relevant. In this perspective, the main goals of this paper are to (i) identify which development approaches of AC are used for designing WSN middleware system architectures that allow the network self-management and (ii) to find out which interactions and behavior can be automated in WSN components, taking into consideration the hardware and software limitations of these networks.

For this purpose, we have conducted a systematic literature review (SLR) [38] to accomplish a methodological, fair analysis about this subject in the literature. In recent years, SLRs have been used for presenting the state of the art regarding a subject topic in a comprehensive, nonbiased way and for identifying interesting and important research opportunities for further investigations.

The rest of this paper is organized as follows. Section 2 describes the conducted SLR. Section 3 discusses the obtained results. Section 4 presents the conclusions of the paper.

2. Systematic Literature Review

This study has been carefully planned as a systematic literature review based on a rigorous methodological framework previously introduced by [38], which provides a set of well-defined steps carried out in accordance with a predefined protocol. This rigorous methodology can be viewed as the main point that differentiates a systematic procedure from a simple, traditional literature review as it seeks to avoid the maximum of bias throughout the process, thus providing scientific value for the obtained findings.

SLRs are means of evaluating and interpreting available relevant research to particular research questions, topic area, or phenomenon of interest, thus aiming to present a fair evaluation of a research topic. A SLR is typically divided into three basic steps: (1) planning, which defines the research questions to be answered, the search strategy to be adopted, the selection criteria, and the data extraction and synthesis methods to be used, thus yielding a protocol that will guide the conduction of the whole process; (2) conduction, in which the primary studies are identified, selected, and evaluated according to the previously established protocol,

Table 1: Final selected sources used in the search stage of this review.

Source	Type	URL
ACM Digital Library	Digital library	http://dl.acm.org/dl.cfm
IEEE Xplore	Digital library	http://ieeexplore.ieee.org/Xplore/home.jsp
ScienceDirect	Digital library	http://www.sciencedirect.com
SpringerLink	Digital library	http://link.springer.com/
Scopus	Digital library	http://www.scopus.com/
ISI Web of Science	Digital library	http://www.webofknowledge.com
Sensors	Digital library	http://www.mdpi.com/journal/sensors
Google Scholar	Search Engine	http://scholar.google.com

Table 2: Search terms used in the online searches.

	Group 1	Group 2	Group 3
Term 1	Autonomic	Wireless sensor_network	Design project
Term 2	Self-adaptive	Sensor network	Design model
Term 3	Self-adapt	WSN	Architecture
Term 4	Self-adaptation	WSAN	Architecture-based
Term 5	Self-adapted	Wireless sensor and actuator network	Framework
Term 6	Self-management	Wireless ad hoc network	Middleware
Term 7	Autonomous	Wireless actuator network	Routing
Term 8			Clustering
Term 9			Data aggregation
Term 10			Data dissemination

and (3) reporting (or analysis), which aggregates extracted information from the relevant primary studies considering the research questions and outlines conclusions from them.

SLRs have been recently viewed as a useful way for dealing with research evidences, thus making it possible to systematically identify, select, analyze, and aggregate them for providing knowledge about a given research topic. Furthermore, they have been commonly used for synthesizing existing work from the literature in a comprehensive and nonbiased way and for identifying research challenges and opportunities in the state of the art regarding the research subject.

The following subsections detail the application of each of these three steps to this SLR.

2.1. Planning. In this phase, the goals and protocol of the SLR were defined. This protocol consists of a predetermined plan that describes the research questions and the search strategy adopted and establishes the selection criteria and the data extraction and synthesis methods.

2.1.1. Research Questions. As the first and most critical step of the tasks performed in our systematic literature review, we translate the goals of our review into research questions. They will be used to find primary studies to understand and summarize evidences about the application of AC principles to WSN in order to optimize network resources. In this context, the following research questions (RQ) were proposed.

RQ1: Which interactions and behavior can be automated in WSN components?

RQ2: Which model/programming/design/developing approach can be applied to provide autonomic behavior to middleware systems for wireless sensor networks?

2.1.2. Search Strategy. At this stage, we perform a search in online digital libraries with a manual compilation of results in order to retrieve all the literature relevant to answer the above specified research questions. For achieving this objective, we specified the sources that can provide the most recent relevant studies for our review and decided how to search in those sources. In order to determine our search sources, we first examined all the online digital libraries and selected the relevant libraries with significant WSNs publications (see Table 1).

After completing the list of sources, we moved on to defining search terms as well as the procedure for searching papers in the online digital libraries. To create our search strings, we first selected multiple key words from our previously defined research questions and then we formed three groups of search terms, as shown in Table 2. Each group contains search terms that are either synonyms (different forms of the same word) or terms that have similar or related semantic meaning within the field. Group 1 and Group 2 encompass the terms "autonomic" and "wireless sensor network," respectively, and its synonyms. Group 3 contains the set of terms used to

```
Query strings-(Group 1) AND (Group 2) AND (Group 3):
TITLE-ABS-KEY
(
    (autonomic OR self-adaptive OR self-adapt OR self-adaptation OR
    self-adapted OR self-management OR autonomous) AND
    ("wireless sensor network" OR "sensor network" OR WSN OR WSAN OR
    "wireless sensor and actuator network" OR "wireless ad-hoc
    network" OR "wireless actuator network") AND
    ("design project" OR "design model" OR "architecture" OR
    "architecture-based" OR framework OR middleware OR routing OR
    clustering OR "data aggregation" OR "data dissemination")
)
```

Algorithm 1

tune our search aiming to find primary studies that explore solutions in a design/architectural level in order to promote autonomic capacities onto WSN.

Most online digital libraries provide advanced search options that allow users to enter Boolean search strings. We fully exploited this feature to construct the search strings used to query each digital library. We defined one search string to search for studies related to research questions defined in this review. We combined the terms of Groups 1, 2, and 3. The general form of the search string is shown in Algorithm 1.

2.1.3. Inclusion and Exclusion Criteria. The purpose of this step is to progressively narrow down the number of articles found in the search stage to an appropriate collection of high quality articles that is thematically relevant for answering the research questions. To complete this task, we eliminate the studies that are not thematically relevant to the scope of this paper. The selection criteria presented in this section involve inclusion criteria and quality screening criteria. The criteria can be further defined as a three-stage process.

Abstract Inclusion Criteria Screening. In this stage, we eliminate some articles that are found in the search phase based on the information provided in the abstract. Articles are kept for further processing if the abstract satisfies one key inclusion criterion; that is, the paper must discuss autonomic features in the context of wireless sensor networks. For the papers with little information in the abstract, we temporarily keep them in the list to be processed in the next stage. Note that, at this stage, we do not consider the quality of the papers.

Full-Text Inclusion Criteria Screening. In this stage, we further eliminate the articles which fail to address the search terms (presented in Table 2) in autonomic wireless sensor network. This means that those papers, despite having the strings in the abstract, only represent minor aspects of the paper.

Full-Text Quality Screening. In this stage, the remaining articles underwent a quality screening where we eliminate studies that do not meet the following quality criteria (QC).

QC1: Is there an overview of the state of the art (related works) and was the rationale for given research clearly justified?

QC2: Is the proposed study addressing the self-* properties of AC?

QC3: Is the proposed study using any AC technique to provide dynamic behavior?

QC4: Are evaluation/validation thoroughly analyzed and explained and do the results of tests strongly support the ideas of autonomic behavior?

2.1.4. Data Extraction and Analysis. The goal of the data collection process is to gather the necessary data to answer research questions in a credible way depending on the quality of data. To ensure data quality, we further set the following criteria:

(1) the works which are published in reliable computer science venues (peer-reviewed conference, peer-reviewed journal, or computer science/engineering organization);

(2) the language for publication which must be in English;

(3) the works which are published during the period of 2000–2014.

After the above process is completed, the extracted data is processed to draw out key themes as part of the synthesis stage of the review. The data that are extracted from each study are detailed in Table 3.

All the findings will be presented and discussed in later sections.

2.2. Conduction. In this phase, the primary studies were searched, selected, and evaluated according to the previously established protocol, thus resulting in a set of possibly relevant studies for the SLR. During the search process, the automated search of primary studies was performed over the selected electronic databases (see Table 1) by searching for all

Table 3: Items and descriptions of the data extraction form.

Items	Descriptions
Title	The title of the primary study
Year	The year when the primary study was published
Source	The conference, journal, or book where the primary study was published
Development approach	Which approaches the selected study uses
Management approach	If the study uses a centralized or distributed solution
Addressed requirements	If the proposed solution considers specific features of WSN
Addressed self-CHOP	Which AC properties are addressed by selected study
Adaptation time	If the study applies adaptation techniques at design or runtime
Topology of network	If the study considers a flat or hierarchic WSN topology
WSN platform	Which platform of WSN was used in the study
Advantages	Benefits of the use of the proposed solution
Limitations	Limitations of the proposed solution
Validation	How the proposed solution was validated

primary studies that matched the adapted search string. The automated search was limited to title, abstract, and keywords fields.

As depicted in Figure 1, 933 studies were retrieved from the electronic databases and 233 of them were initially selected based on title, abstract, and keywords against the selection (inclusion/exclusion) criteria. From the initial set of 233 studies, the selection criteria were applied during the full reading, thus resulting in a set of 62 primary studies. The remaining articles underwent a quality screening where we eliminated 37 studies that did not meet the quality criteria; thus 25 primary studies were considered as relevant to this SLR and then selected for data extraction.

Figure 1 depicts these steps for selecting the relevant primary studies. Table 4 shows the list of studies that were considered as relevant for data extraction. Figure 2 shows the number of selected studies classified per year of publication. Despite the SLR retrieved studies from 2000, the most relevant contributions in terms of updating the state of the art about autonomic WSNs come from 2007.

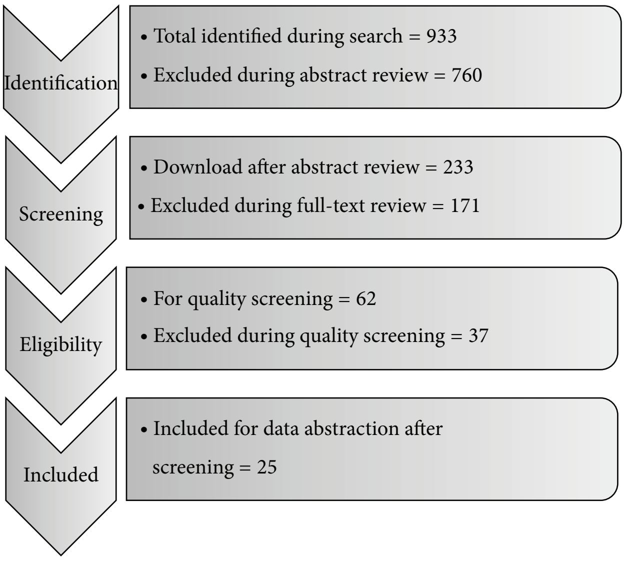

Identification
• Total identified during search = 933
• Excluded during abstract review = 760
Screening
• Download after abstract review = 233
• Excluded during full-text review = 171
Eligibility
• For quality screening = 62
• Excluded during quality screening = 37
Included
• Included for data abstraction after screening = 25

Figure 1: Procedure for selecting relevant primary studies.

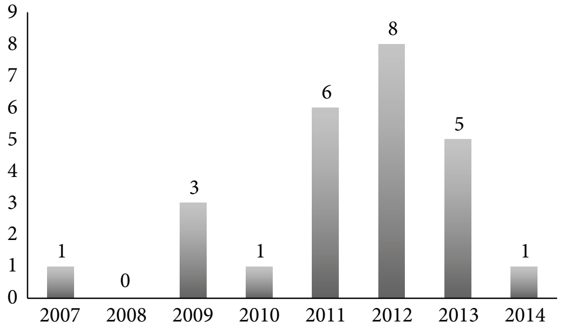


Figure 2: Number of selected studies per year.

2.3. Reporting. This phase presents the results of the SLR according to the defined research questions in light of the selected studies.

2.3.1. Research Question 1 (RQ1). This research question is related to what can be automated in wireless sensor networks. We noticed that most of the selected studies are interested in a lightweight, autonomic behavior for WSNs in order to (i) adapt the network to dynamic environments and unpredictable events, (ii) save energy, extending the network lifetime, (iii) provide scalability of network, (iv) provide reliability of sensing data, and (v) provide independence between the management functions of applications and network configuration. The selected studies address the aforementioned concerns according to the self-* properties of AC.

Selected studies, S1, S2, S4, S5, S12, S13, S16, and S22, focus on propose solutions to address the *self-configuration* property. Sensor nodes reconfigure and adapt their behaviors of networking and sensing by altering parameter values dynamically according to the changing conditions and states of the network. Some examples of network configuration adopted in S22 are decreasing the node sensing duty cycle if the monitored phenomenon has no significant changes in a period of time and reducing radio transmission power to shorten the communication range if the residual node energy has dropped to a critical level.

S13, S21, S22, S23, S24, and S25 address the *self-healing* property. This property may be considered an essential characteristic that a sensor network should incorporate for

Table 4: Selected primary studies.

ID	Title	Year	Reference
S1	Design of a Generic Management System for Wireless Sensor Networks	2014	[11]
S2	Smart Policy Generating Mechanism for Policy Driven Self-Management in Wireless Sensor Networks	2013	[12]
S3	A QoS-Driven Self-Adaptive Architecture for Wireless Sensor Networks	2013	[13]
S4	Using Dynamic Software Variability to Manage Wireless Sensor and Actuator Networks	2013	[14]
S5	DISON: A Self-organizing Network Management Framework for Wireless Sensor Networks	2013	[15]
S6	Autonomous Configuration of Spatially Aware Sensor Services in Service Oriented WSNs	2013	[16]
S7	A Novel Wireless Sensor and Actor Network Framework for Autonomous Monitoring and Maintenance of Lifeline Infrastructures	2012	[17]
S8	Constraint-Based Self-adaptation of Wireless Sensor Networks	2012	[18]
S9	Framework for a Self-managed Wireless Sensor Cloud for Critical Event Management	2012	[19]
S10	Autonomous Sensor Network Architecture Model	2012	[20]
S11	Developing Wireless Sensor Network Applications Based on a Function Block Programming Abstraction	2012	[21]
S12	Autonomous Sensor Networks for Process Monitoring and Automation	2012	[22]
S13	An Autonomic Plane for Wireless Body Sensor Networks	2012	[23]
S14	Framework for Distributed Policy-Based Management in Wireless Sensor Networks to Support Autonomic Behavior	2012	[24]
S15	Autonomic Role and Mission Allocation Framework for Wireless Sensor Networks	2011	[25]
S16	Towards Aware, Adaptive and Autonomic Sensor-Actuator Networks	2011	[26]
S17	Autonomic Computing Driven by Feature Models and Architecture in FamiWare	2011	[27]
S18	Autonomous Decentralized Mechanism of Structure Formation Adapting to Network Conditions	2011	[28]
S19	Swarm Behavior Control of Mobile Multi-Robots with Wireless Sensor Networks	2011	[29]
S20	Middleware Support for a Self-Configurable Wireless Sensor Network	2011	[30]
S21	Starfish: Policy Driven Self-Management in Wireless Sensor Networks	2010	[31]
S22	Autonomic Networking in Wireless Sensor Networks	2009	[32]
S23	Secure Self-Adaptive Framework for Distributed Smart Home Sensor Network	2009	[33]
S24	Agilla: A Mobile Agent Middleware for Self-Adaptive Wireless Sensor Networks	2009	[34]
S25	A Survey on the Applicability of Trust Management Systems for Wireless Sensor Networks	2007	[35]

assuring its reliability and correctness. Fault tolerance is a common feature addressed in this context. For example, as noted by S22, to lessen the impact of faulty nodes, sensor nodes surrounding the phenomenon area generally regroup among themselves in order to maintain the reliability and consistence of network connectivity and sensing coverage. After the faulty neighbor is detected, a node will choose a new neighbor to route to. In S13, a filtering task is interposed between the sensing and the processing in order to guarantee the quality of raw data and to avoid that a data corruption affects the entire application correctness.

Regarding the *self-optimization* property, in order to extend the WSN operating lifetime, data transmission should be avoided and sensing data must be processed, as is done in S2, S3, S4, S5, S10, S12, and S13. For example, S10 proposes a basic filtering process to recognize that there is unnecessary information in the raw sensor data able to attend application requirements. Thus, it is possible to reduce the number of transmitted data packets and consequently reduce the total energy usage at sensor nodes.

Finally, regarding the *self-protection* property, an encryption process could be conceived in order to encrypt data coming from sensor nodes, if necessary. For S13, the privacy of information transmitted in a sensor network is one of the highest priority goals regarding this AC property. In order to provide self-protection in WSN applications, S23 proposes a secure communication method applied to smart homes. This method is a nature inspired framework based on mimicking ant's behavior. Soleman and Payadeh [39] propose an autonomic mechanism to detect attacks in WSN. The protection mechanism depends on detecting the abnormal behavior in the network. This mechanism is located in a base station. All the cluster heads send their data directly to the base station. The operation of the mechanism is similar to the work of the brain; the brain receives data from the whole body and detects abnormal behavior. For Dan Wang [40] a sensor network is k-self-protected if each sensor (active or inactive) is covered by at least k-1 active sensors. In [40] a 2-self-protection only is used.

2.3.2. Research Question 2 (RQ2). This research question is related to which development approaches can be applied to provide autonomic behavior to wireless sensor networks applications. From the analyzed studies, we have identified that selected studies use the following approaches:

Table 5: Policy structure proposed by S14.

Policy structure	
[ID] policy ID	3 bytes
[If] policy condition	[If]
[Then] do policy action	3 bytes
[End] end policy execution	1 byte
[Next] execute next policy ID	3 bytes

(i) policy-based reasoning approach; (ii) context-based reasoning approach; (iii) feedback control loops; (iv) mobile agents; (v) model transformation and code generation. The analysis of every approach is shown below.

Studies S1, S2, S5, S6, S10, S14, and S21 rely on *policy-based reasoning (PBR)* approaches. As stated in S14, a general way of implementing autonomic behavior in distributed systems is through the use of policies. A policy is a constraint on the system behaviors that can be expressed using natural languages or mathematical notations. Policy-based systems use many existing expressive languages for specifying policies, but, due to resource constraints, they are not appropriate for wireless sensor networks. Table 5 depicts a policy structure specific for WSN [21]. Typically a policy in WSN is specified in terms of tasks to monitor events, verify conditions, and trigger actions whenever predefined events are detected by the monitoring task.

In Table 5, (i) [ID] policy ID is used throughout the WSN to locate any particular policy. A policy identificator consists of only two parts which are *Event ID* and sequence number of policy (*SeqNo*). *Event ID* is 2 bytes long. The first byte represents the event category such as T, Temperature = 1; the second byte is a hexadecimal number representing the sequence number of possible events in the sensor. *SeqNo* is 1 byte long representing policy sequence within the chain of applicable policies to the *Event ID*. *Event ID* and *SeqNo* are sensor dependent information and can be locally accessed from the sensor. Thus the sensor can identify the policy ID locally without the need to reach out to any other sensors; (ii) [If] policy condition is a Boolean expression based upon the data provided by the local sensor system and static or dynamic data provided by the triggered policy; whenever an event occurs, one or more conditions are checked as true, and if all of them pass, a corresponding action must be executed; (iii) [Then] policy action describes the desired action number (ID) to be executed when the IF condition is true; (iv) [End] end policy execution indicates the end of the policy execution if the condition is false; otherwise the policy execution will move to the next policy in the chain; (v) [Next] next policy ID contains the key for the next policy in the chain of applicable polices.

Following the *context-based reasoning (CBR)* approach, S4, S5, S9, S13, S16, S18, and S23 consider as meaningful context any information that affects node's operation. Context information constitutes an important source of data for systems that have to react dynamically to changes in the environment or to new context conditions. For instance, S5 predefines the following context formats and stores them in a central database: `[CONTEXT ID] [INFORMATION TYPE] [INFORMATION ID] [INFORMATION VALUE]`.

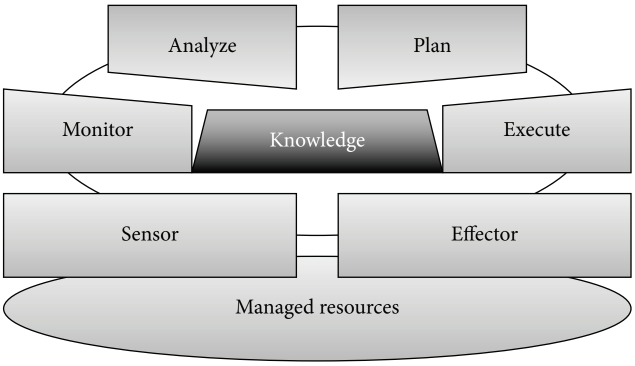


Figure 3: Closed feedback control loop in autonomic systems.

`CONTEXT ID` is the unique identifier of the context; `INFORMATION TYPE` describes the source of the raw information; `INFORMATION ID` represents the identifier of each specific information such as the sensing capabilities and the residual energy; and `INFORMATION VALUE` is the value of that specific information in bits. Once obtained the context is possible to apply a reasoning process in order to react to dynamic changes of network. On the other hand, S16 uses fuzzy logic and machine learning techniques as a reasoning process for this stage. Finally, S1 applies a hybrid approach relying on policy-based and context-based approach.

The feedback control loop (FBL) approach (S1, S3, S5, S7, S15, and S20) is commonly presented in AC systems and most of them use four steps (monitoring, analysis, planning, and execution) as illustrated in Figure 3.

Generally, WSN middleware systems that apply this approach use hierarchical networks (Figure 4) typically defined with three (or more) levels: sensor nodes level, cluster head level, and a base station level. This makes the system able to provide quick adaptation to multiple context parameter changes.

Mobile agents (MA) approach (S11, S19, and S24) provides a programming model in which applications consist of evolving communities of agents that share a WSN. Agents can dynamically enter and exit a network and can autonomously clone and migrate themselves in response to changes in the environment. Users inject mobile agents that spread across nodes performing application-specific tasks. Each agent is autonomous, allowing multiple applications to share a network.

Model transformation/code generation (MT/CG) approaches define an automatic process to derive different middleware configurations depending on the hardware and software of the deployed WSN. This process uses techniques such as model-driven development (MDD) [41] and software product line (SPL) [42]. Model transformation and automatic code generation (S8, S12, S17, and S20) are used to create the concrete system from the model. This approach allows nonexperts to develop WSN systems and to provide the needed mechanisms to adapt the network to dynamic environments and unpredictable events.

Table 6: Summarization of primary studies relevant data.

Reference	Development approach	Management approach	Type of solution	Addressed self-properties	Adaptation Time	Topology of WSN	Evaluation
S1	PBR and FBL	Hybrid	Framework	Self-configuration	Runtime	Flat/hierarchic	TinyOS/TelosB motes
S2	PBR	Hybrid	Framework	Self-configuration Self-optimization	Design/runtime	Hierarchic	Simulated in Contiki OS
S3	FBL	Hybrid	Middleware	Self-optimization	Runtime	Flat/hierarchic	Simulated in Avrora
S4	CBR	Hybrid	Middleware	Self-configuration Self-optimization	Design	Flat/hierarchic	No
S5	PBR and FBL	Hybrid	Framework	Self-configuration Self-optimization	Runtime	Hierarchic	No
S6	PBR	Centralized, distributed	Middleware, Framework	Self-configuration	Runtime	Flat	Simuled in CORE and EMAN
S7	FBL	Hybrid	Framework	Self-configuration Self-optimization Self-healing	Runtime	Hierarchic	No
S8	MT/CG	Hybrid	Middleware	Self-configuration	Design	Hierarchic	Simulated on RecosQos
S9	CBR	Distributed	Framework	Self-configuration	Runtime	Hierarchic	Libelium Waspmotes
S10	PBR	Distributed	Middleware	Self-optimization	Design	Hierarchic	No
S11	MA	Distributed	Middleware	Self-configuration	Runtime	Flat/hierarchic	—
S12	MT/CG	Hybrid	Middleware	Self-configuration Self-optimization	Design Runtime	Hierarchic	ContikiOS/Atmel AVR
S13	CBR	Distributed	Framework	Self-configuration Self-healing Self-optimization Self-protection	Design/runtime	Hierarchic	TinyOS
S14	PBR	Hybrid	Framework	Self-configuration	Design/runtime	Flat/hierarchic	Finger/Finger 2
S15	FBL	Distributed	Framework	Self-configuration	Runtime	Flat/hierarchic	TinyOS 2.x, Finger 2
S16	CBR	Distributed	Framework	Self-configuration	Design/runtime	Hierarchic	Simulated
S17	MT/CG	Hybrid	Middleware	Self-configuration	Design/runtime	Flat/hierarchic	TinyOS 2.1.1, TOSSIM
S18	CBR	Distributed	Framework	Self-configuration	Design	Flat/hierarchic	Simulations
S19	MA	Distributed	Framework	Self-configuration	Runtime	Not specific	Simulations
S20	MT/CG	Distributed	Middleware	Self-configuration	Design/runtime	Flat/hierarchic	No
S21	PBR	Distributed	Middleware, Framework	Self-configuration Self-healing	Runtime	Flat/hierarchic	TinyOS
S22	—	Centralized, distributed	—	Self-configuration Self-healing	Runtime	Flat/hierarchic	TinyOS
S23	CBR	Distributed	Framework	Self-healing Self-protection	Runtime	Flat/hierarchic	Simulated in MATLAB
S24	MA	Distributed	Middleware	Self-configuration Self-healing	Design/runtime	Not specific	TinyOS Mica2, Telosb
S25	—	Centralized, distributed	—	Self-healing	Runtime	Flat Hierarchic	—

Issues highlighted of each primary study are summarized in Table 6. The first column (development approach) shows the approach followed for every study. Furthermore, in Figure 5 is depicted the different development approaches used to achieve every AC property.

The management approach column shows techniques used to implement the before-mentioned development approaches. These techniques can be centralized, distributed, or hybrid. In a centralized approach, the control of WSN management is centrally located in the sink node; in a distributed approach, the control of management is fully distributed among the WSN nodes. The sensor nodes apply coordination functions to manage themselves. Most of the selected studies use a hybrid approach, where part of the management functions are performed in the sink nodes and parts are distributed among the sensor nodes.

We also noticed that selected studies implement their proposals at the middleware level (see type of solution column). Middleware frameworks reduce the time and effort in developing WSN applications, by providing an easy way to integrate complex and distributed autonomic services, common programming abstractions, and hiding low-level

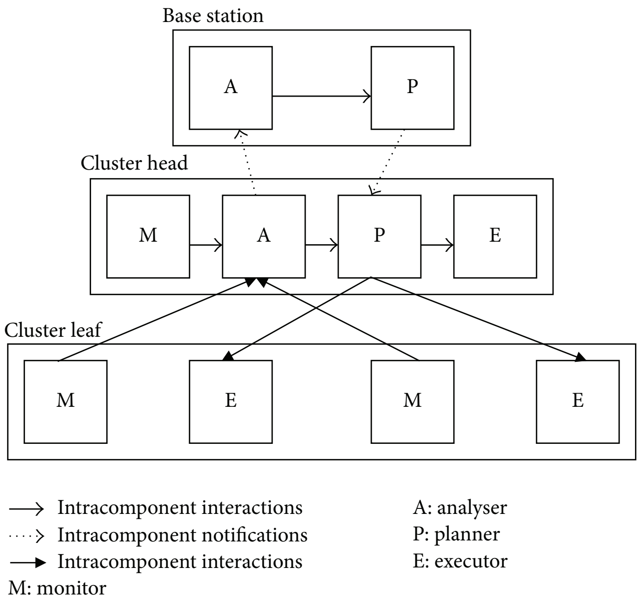


Figure 4: Self-adaptive control loop distributed in three levels of network architecture.

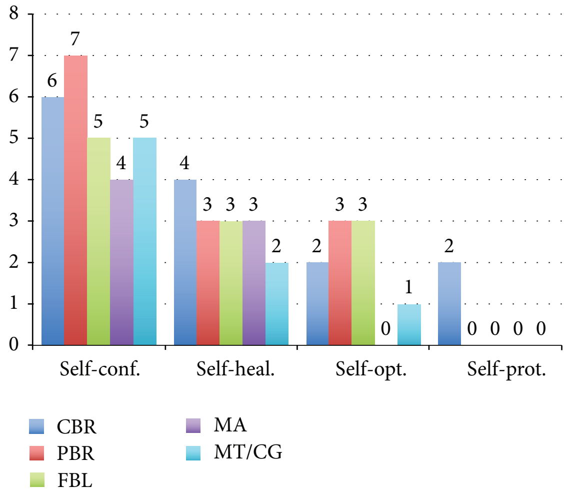


Figure 5: Correlation between the use of AC properties in development approaches.

programming details of different sensor platforms. Thus, developers can devote more time in developing the WSN application requirements and to (re)use the autonomic management components of middleware to configure the self-adaptive behavior of WSN.

In the column named addressed self-properties we noted that the self-configuration, in autonomic WSN middleware systems, is a general-purpose property. However, implementations of self-healing and self-protection are most common in specific WSN applications.

The adaptation time column specifies when the adaptation process occurs. Most of the selected studies execute an adaptation plan at runtime. The MT/CG approach defines the adaptation plan at design time, before the creation of the concrete system. Regarding the WSN topology used in selected studies (topology of WSN column), we noticed that hierarchic topologies are most used in hybrid/distributed approaches.

Finally, the last column shows the evaluation process used for validating the proposals of selected studies. Most of the selected studies simulate their tests using TinyOS-based programs.

3. Discussion

After performing the SLR, we noticed that most works still are in an initial stage of integration of AC principles into WSN systems. The goal of optimizing network resources by using these principles is a relatively new research topic in the WSN field. Dealing with AC in middleware systems for WSN is more complex when compared to their counterparts in traditional middleware systems mainly due to the need of handling limited computing resources of sensor nodes. However, we believe that the selected studies are useful means for addressing autonomic behavior in WSN applications. In the following, we present some challenges and research opportunities that we have identified from the analyzed studies.

As we have mentioned, one of development approaches of AC used to provide autonomic behavior in WSN is context-based reasoning. In this approach, the network adaptation is limited once it is a simple reaction to the current context of WSN. The middleware system receives the network context as input and replies to it following a logic that selects the most appropriate action. The most important activity of this approach is to sense what is happening in the network and if the WSN receives a stimulus, the WSN reacts accordingly. The reactivity feature of context-based middleware systems derives from the fact that they just are able to perceive the environment. So, these systems can react to events that occur in the environment in order to satisfy their design objectives.

The policy-based reasoning approach is widely used in goal-oriented WSN applications. In this approach, the choice of a specific action depends not only on the context of the network but also on how close to the goal each action will bring the system. Systems applying this approach do everything possible in order to achieve the goal of the WSN application. The conventional policy-based systems are generally too heavy to execute in a sensor node. Due to these limitations (memory and CPU constraints), devices in WSN can only store a limited number of policies in their memory and must recycle them when required. This process of loading/unloading policies might create a communication overhead that needs to be handled. Policy-based middleware systems are able to take initiatives towards the satisfaction of specific internal design objectives.

In the mobile agents approach a WSN is seen as a platform which software agents can use in order to perform sensing and/or computing tasks. Agents are highly autonomous and can make adaptation decisions locally based on the changes of the environment. Such decisions take form when the agent decides to migrate or clone itself to neighbor nodes.

This approach enables the creation of self-adaptive, self-organized, and autonomous applications. As network nodes are directly exposed to the environment, agents can quickly detect changes and determine when adaptation is necessary. Therefore, autonomous and localized agents react faster and transmit less data than central adaptation approaches. This makes them suitable for applications in which local decisions significantly reduce the amount of data wirelessly transmitted. The paramount example is tracking of an object (person, fire, or wildlife) as it passes through an area monitored by a WSN. The agents responsible for tracking can migrate along the WSN nodes as the object passes through them, avoiding wasting resources on nodes that are far from the object. On the other hand, the mobile agents approach is not meant for data collection applications that require deployment across the entire WSN. On such cases this approach introduces an unnecessary overhead of agents switching, which requires wireless data transmission and dynamic-runtime memory allocation. One interesting and rather unexplored application of mobile agents in WSN is modelling the behaviors of swarm individuals. Such can be accomplished by rewarding or penalizing the agent's approaches and movements. The effect can then be observed as a whole across the WSN and can turn into a powerful way to study the emergence of swarm behavior.

Model transformation and code generation approaches are traditionally used in software product lines (SPL) in order to create static software systems from a set of software assets. However these techniques can also be employed to create dynamic software able to do self-configuration. In S8 and S17 the authors propose a middleware platform that allows the creation of self-configurable WSN applications. Applications are described using a feature model (FM), which is then translated into code that uses the middleware in order to cope with node heterogeneity. The FM can be extended to cope with multiple scenarios; for instance, it can define a choice between several routing protocols. In that way these models can also be used at runtime to drive the reconfiguration of the middleware for failure recovery and/or self-configuration; this is known as a models@run.time approach.

Using models@run.time has the advantage of keeping the application within a known state described by a model, even when it mutates to cope with changes in the environment. This eases the burden of keeping track of architectural configurations in applications that are scattered across several nodes. Furthermore, by doing the self-configuration in a middleware layer common to all nodes, this approach allows a higher degree of adaptability than others like the mobile agents approach. For instance, it is possible to change the routing protocol in runtime. However, architectural configurations that must take place in every node to ensure compatibility, like changing the routing protocol, require sinks or cluster heads to coordinate adaptation across the multiple nodes. This reduces the locality of the adaptation and incurs in high communication overheads. Also, changing the model on a node is a rare but computing intensive task. The node has to forge a plan detailing the configuration steps to take and then must use a domain modeler to check the correctness of the plan, that is, check if it arrives at a valid state defined in the FM.

In the before-mentioned approaches, adaptation is possible with the explicit or implicit presence of feedback loops. WSN applications with explicit feedback loops define a part of the system that deals with feedback. This part of the system is able to interact, communicate, and coordinate among middleware components. The feedback control loop approach is considered essential for understanding not only the model of adaptation and collaboration, but also the types of adaptive systems. It can be considered the most dynamic adaptive approach. The majority of the selected studies use the MAPE-K model proposed by IBM [37].

In order to extend and implement the aforementioned approaches, system architectures, sitting between the sensing applications and the node operating system, are expected to provide a set of integrated functions for nodes to be self-manageable and self-configurable. Nevertheless, there is still no comprehensive system architecture design that supports these expectations. For the authors in [32], it is unfeasible to predict any node failure, under highly dynamic, remote, and hostile environments of sensing applications. In order to support the management system to take efficient recovery actions and successfully resume from a failure, it is necessary to consider the trade-off between the complexity of fault management functions of system architecture and the resource constraints of sensor nodes.

Figure 6 depicts a summary of main features of autonomic wireless sensor network, gathered from selected studies of this SRL. These features are organized in terms of (i) development approaches used to provide autonomic behavior in WSN, (ii) requirements of autonomic WSN, (iii) self-adaptable/self-manageable features of WSN, according to AC principles, and (iv) techniques used by the development approaches.

4. Conclusion

This paper aimed to present a systematic literature review with the purpose of obtaining the state of the art of approaches, methods, and methodologies whose goal is the use of AC principles in wireless sensor network applications in order to optimize network resources.

Therefore, we defined the SLR protocol and presented the search and the results from this review. As a result, we have found that selected studies address AC principles based on its self-* properties: self-configuration, self-healing, self-optimization, and self-protection. Also, we have found studies with different development approaches: context-based reasoning, policy-based reasoning, feedback control loop, mobile agents, and model transformation and code generation.

More than the half of the selected studies propose solutions for self-configuration property, but few of them address self-protections and self-optimization properties. Most of the feedback control loop solutions use context-based reasoning for monitoring process and policy-based reasoning to execution an action plan. The selected studies that implemented

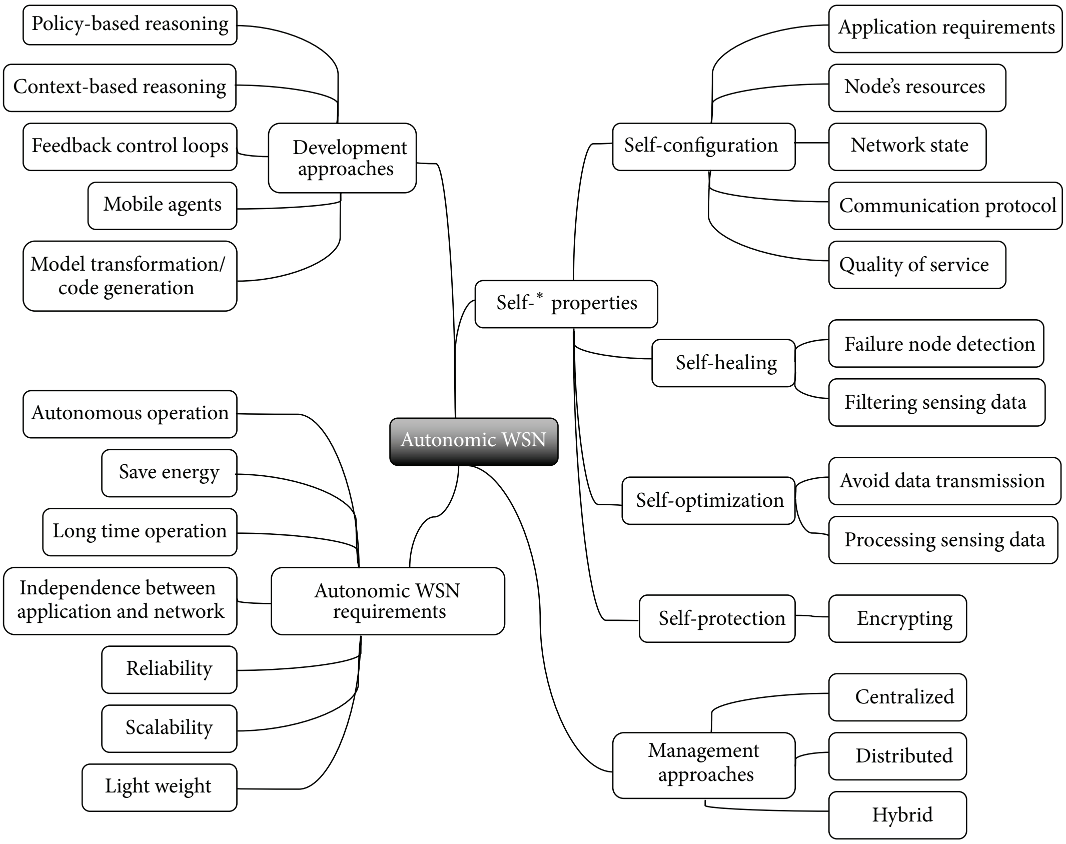


FIGURE 6: Autonomic wireless sensor network characteristics.

a combination of these development techniques were better able to handle the four self-* properties.

Deploying WSN in insecure environments and using the wireless transmission and the node limited sources keeping the security of data and control information is an important open issue for WSN. We noticed a lack of solutions for self-protection in WSN. Some security requirements of self-protection are authentication, integrity, and confidentiality. It is necessary to propose solutions to defend the sensor network against correlated problems arising from malicious attacks or cascading failures that remain uncorrected by self-healing measures and to propose mechanisms to anticipate problems based on early reports from sensors and taking steps to avoid or mitigate them. One of the main challenges is to define mechanisms able to detect many types of unknown and known attacks.

Regarding evaluation methods, the selected studies are usually evaluated using simulators. Running real experiments on a testbed is costly and difficult. Also, execution of tests is largely compromised since many factors affect experimental results at the same time. Moreover, running real experiments are always time consuming. WSN simulators allow isolating factors. However, the fundamental trade-off is precision and necessity of details versus scalability and performance.

Finally, there is a lack of well-defined architecture design that supports the autonomy of sensor networking.

Summarizing, most of the works that we have found in the literature that apply approach to provide autonomic behavior to the WSNs are just in preliminary stages and they have still some open challenges. However, their proposals seem very adequate to tackle some aspects of the autonomic WSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Professors Flávia Delicato and Paulo Pires are CNPq Fellows. This work was partially supported by the Brazilian funding agencies CAPES, CNPq, and FAPERJ and by the European INTER-TRUST FP7-317731 Project and the Spanish TIN-2012-34840, FamiWare P09-TIC-5231, and MAGIC P12-TIC1814 Projects.

References

[1] V. Potdar, A. Sharif, and E. Chang, "Wireless sensor networks: a survey," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 636–641, May 2009.

[2] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2007.

[3] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–29, 2005.

[4] M.-M. Wang, J.-N. Cao, J. Li, and S. K. Dasi, "Middleware for wireless sensor networks: a survey," *Journal of Computer Science and Technology*, vol. 23, no. 3, pp. 305–326, 2008.

[5] S. Hadim and N. Mohamed, "Middleware: middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, pp. 1–23, 2006.

[6] A. Rahman, "Middleware for wireless sensor networks: challenges and approaches," in *Proceedings of the Seminar on Internet Working*, Helsinki University of Technology, Espoo, Finland, April 2009.

[7] T. Liu and M. Martonosi, "Impala: a middleware system for managing autonomic, parallel sensor systems," *Parallel Sensor Systems*, vol. 38, no. 10, pp. 107–118, 2003.

[8] P. Costa, G. Coulson, R. Gold et al., "The RUNES middleware for networked embedded systems and its application in a disaster management scenario," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*, pp. 69–78, March 2007.

[9] K. K. Khedo and R. K. Subramanian, "Service-oriented component-based middleware architecture for wireless sensor networks," *International Journal of Computer Science and Network Security*, vol. 9, no. 3, pp. 174–182, 2009.

[10] F. C. Delicato, P. F. Pires, L. Pirmez, and L. F. R. C. Carmo, "A flexible middleware system for wireless sensor networks," in *Middleware 2003: ACM/IFIP/USENIX International Middleware Conference Rio de Janeiro, Brazil, June 16–20, 2003 Proceedings*, vol. 2672 of *Lecture Notes in Computer Science*, pp. 474–492, 2003.

[11] T. M. Cao, B. Bellata, and M. Oliver, "Design of a generic management system for wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 16–35, 2014.

[12] S. Sun, B. Zeng, and J. Liu, "Smart policy generating mechanism for policy driven self-management in wireless sensor networks," *Sensors & Transducers*, vol. 154, no. 7, pp. 9–14, 2013.

[13] A. Jemal and R. Ben Halima, "A QoS-driven self-adaptive architecture for wireless sensor networks," in *Proceedings of the 22nd IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '13)*, pp. 125–130, Hammamet, Tunisia, June 2013.

[14] M. L. Mouronte, O. Ortiz, A. Belen Garcia, and R. Capilla, "Using dynamic software variability to manage wireless sensor and actuator networks," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM '13)*, pp. 1171–1174, May 2013.

[15] T. Minh, B. Bellalta, and M. Oliver, "DISON: a self-organizing network management framework for wireless sensor networks," in *Ad Hoc Networks*, vol. 111 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 149–163, Springer, Berlin, Germany, 2013.

[16] S. Y. Shah, B. Szymanski, P. Zerfos, C. Bisdikian, C. Gibson, and D. Harries, "Autonomous configuration of spatially aware sensor services in service oriented WSNs," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '13)*, pp. 312–314, IEEE, San Diego, Calif, USA, March 2013.

[17] M. Imran, M. A. Alnuem, W. Alsalih, and M. Younis, "A novel wireless sensor and actor network framework for autonomous monitoring and maintenance of lifeline infrastructures," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 6484–6488, Ottawa, Canada, June 2012.

[18] N. Gamez, D. Romero, L. Fuentes, R. Rouvoy, and L. Duchien, "Constraint-based self-adaptation of wireless sensor networks," in *Proceedings of the 2nd International Workshop on Adaptive Services for the Future Internet and 6th International Workshop on Web APIs and Service Mashups (WAS4FI-Mashups '12)*, pp. 20–27, September 2012.

[19] N. Nair, P. Morrow, and G. Parr, "Framework for a self-managed wireless sensor cloud for critical event management," in *Sensor Systems and Software*, vol. 102 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 15–29, 2012.

[20] A. Tóth and F. Vajda, "Autonomous sensor network architecture model," in *Information and Communication Technologies*, vol. 7479 of *Lecture Notes in Computer Science*, pp. 298–308, Springer, Berlin, Germany, 2012.

[21] F. Kerasiotis, C. Koulamas, and G. Papadopoulos, "Developing wireless sensor network applications based on a function block programming abstraction," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '12)*, pp. 372–377, March 2012.

[22] G. Balakrishnan and S. S. Hiremath, "Autonomous sensor networks for process monitoring and automation," in *Proceedings of the 10th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI '12)*, pp. 47–52, Herl'any, Slovakia, January 2012.

[23] G. Fortino, S. Galzarano, and A. Liotta, "An autonomic plane for wireless body sensor networks," in *Proceedings of the International Conference on Communications and Network*, pp. 94–98, January 2012.

[24] N. Qwasmi and R. Liscano, "Framework for distributed policy-based management in wireless sensor networks to support autonomic behavior," *Procedia Computer Science*, vol. 10, pp. 232–239, 2012.

[25] T. Bourdenas, K. Tei, S. Honiden, and M. Sloman, "Autonomic role and mission allocation framework for wireless sensor networks," in *Proceedings of the 5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '11)*, pp. 61–70, Ann Arbor, Mich, USA, October 2011.

[26] M. ElGammal and M. Eltoweissy, "Towards aware, adaptive and autonomic sensor-actuator networks," in *Proceedings of the 5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '11)*, pp. 210–211, October 2011.

[27] N. Gamez, L. Fuentes, and M. Aragüez, "Autonomic computing driven by feature models and architecture in famiware," in *Software Architecture: Proceedings of 5th European Conference, ECSA 2011, Essen, Germany, September 13–16, 2011*, vol. 6903 of *Lecture Notes in Computer Science*, pp. 164–179, 2011.

[28] C. Takano, M. Aida, M. Murata, and M. Imase, "Autonomous decentralized mechanism of structure formation adapting to network conditions," in *Proceedings of the 11th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT '11)*, pp. 524–531, July 2011.

[29] W. Li and W. Shen, "Swarm behavior control of mobile multi-robots with wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1398–1407, 2011.

[30] M. Götz, A. Rettberg, and I. Podolski, "Middleware support for a self-configurable wireless sensor network," in *Proceedings of the 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW '11)*, pp. 143–151, IEEE, Newport Beach, Calif, USA, March 2011.

[31] T. Bourdenas and M. Sloman, "Starfish: policy driven self-management in wireless sensor networks," in *Proceedings of the ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '10)*, pp. 75–83, May 2010.

[32] D. Garlan, B. Schmerl, and S. Cheng, *Autonomic Computing and Networking*, 2009.

[33] R. Muraleedharan and L. A. Osadciw, "Secure self-adaptive framework for distributed smart home sensor network," in *Proceedings of the 43rd Asilomar Conference on Signals, Systems and Computers*, pp. 284–287, IEEE, November 2009.

[34] C.-L. Fok, G.-C. Roman, and C. Lu, "Agilla: a mobile agent middleware for self-adaptive wireless sensor networks," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 3, article 16, 2009.

[35] M. Fernandez-Gago, "A survey on the applicability of trust management systems for wireless sensor networks," in *Proceedings of the 3rd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SECPerU '07)*, pp. 25–30, 2007.

[36] M. Salehie and L. Tahvildari, "Self-adaptive software: landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 2, article 14, 2009.

[37] R. Lanyon-Hogg, "Front cover a Practical Guide IBM Autonomic mic".

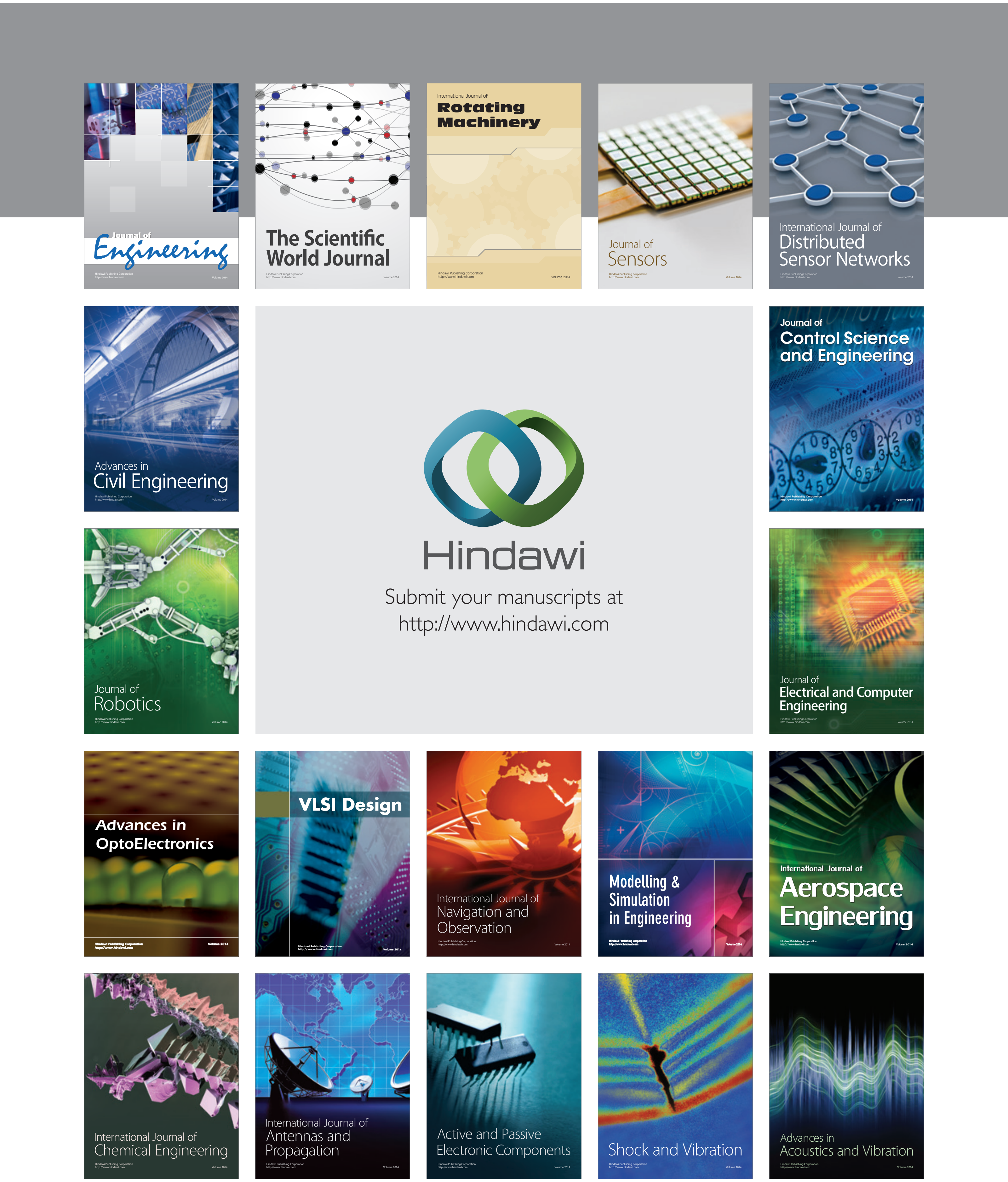
[38] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering version 2.3," Tech. Rep., Keele University and University of Durham, 2007.

[39] H. Soleman and A. Payandeh, "Self-protection mechanism for wireless sensor networks," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 6, no. 3, 2014.

[40] D. Wang, Q. Zhang, and J. Liu, "The self-protection problem in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 4, article 20, 2007.

[41] Model Driven Architecture, http://www.omg.org/mda.

[42] K. Pohl, G. Böckle, and F. J. van der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer, Berlin, Germany, 2005.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration