MDE-Based Languages for Wireless Sensor Networks Modeling: A Systematic Mapping Study

Fatima Essaadi, Yann Ben Maissa and Mohammed Dahchour

Abstract Wireless Sensor Networks (WSNs) are ubiquitous systems of small devices equipped with sensors that collaborate to sense physical quantities in an area. However, the design constraints, the behavior requirements and the error prone nature, make the development of WSNs and their deployment an extremely challenging task. The Model Driven Engineering (MDE) approach helps tackling these issues by using models and automatic transformation to generate code or analyze WSNs against their requirements. In this paper, we propose a systematic mapping study which presents the existing WSNs MDE-based modeling languages. We surveyed a total of 1852 papers from which we selected 21 languages satisfying 7 selection criteria. We analyze these languages according to 5 rigorous research questions and 12 comparative criteria. Then we provide a precise view on the existing languages and their weaknesses mainly regarding mobility and data fusion. Finally, we propose research directions and recommendations for aspiring languages developers.

Keywords Wireless sensor networks · Model driven engineering · Domain specific modeling language · Model transformation · Systematic mapping study

1 Introduction

Context and Problem. Wireless Sensor Networks (WSNs) have emerged as one of the most promising technologies for the future [1]. WSNs have achieved great results in several domains such as health monitoring. Nevertheless, the application development in such a domain is a complicated process for the following reasons:

F. Essaadi (✉) · Y. Ben Maissa · M. Dahchour
Telecommunication Systems, Networks and Services Lab,
National Institute of Posts and Telecommunications, Rabat, Morocco
e-mail: essaadi@inpt.ac.ma

Y. Ben Maissa
e-mail: benmaissa@inpt.ac.ma

M. Dahchour
e-mail: dahchour@inpt.ac.ma

R. El-Azouzi et al. (eds.), *Advances in Ubiquitous Networking 2*, Lecture Notes
in Electrical Engineering 397, DOI 10.1007/978-981-10-1627-1_26

First, once the network deployment is carried out, it will be very costly and time-consuming to make modifications if the network goes down.

Second, the sensor node, as an embedded system, is characterized by the hard coupling between the software and the hardware [2]. As a result, expert developers must deal with low-level implementation details in their applications development which results in highly platform-dependent designs. This makes the designs a hard task to maintain, modify and reuse.

Third, the deployment of WSNs is strongly influenced by several constraints which can be categorized in application and design constraints. While the application constraints are application dependent, the limited energy is the most important design constraint [3]. The energy consumption has the higher priority even more than quality of service [3]. All these constraints make WSNs application development complicated and error prone.

In order to tackle these difficulties, the current proposals insist on adopting a Software Engineering Paradigm to support the application life cycle development. The Model Driven Engineering (MDE) approach is proposed for this aim. MDE is a software engineering approach which is based on the principle of *Everything is model* [4]. The most relevant advantages presented by MDE are:

First, it raises the abstraction level and hides low-level implementation details. So, the domain experts can focus on the domain problems instead of focusing on technological aspects (e.g., target platforms).

Second, MDE is less error-prone, because it allows analysis in the early stage of life cycle development which reveal errors (e.g., missed performance requirements) before network deployment. Then the developers can customize their designs in regard to detected errors.

Third, MDE introduces the concept of Model Transformation which incorporates a set of rules to apply sequentially over models usually in order to produce the system code or the analysis model. Thus the code generation becomes an easy task regarding its complexity in traditional programming languages.

Contribution. In this paper, we propose a Systematic Mapping Study (SMS) to study the existing MDE-based languages. A SMS differs from a Systematic Literature Review (SLR). While in SLR, we deeply review existing primary studies and summarize their methodologies and results, a SMS is a defined method to provide a *precise* classification scheme and structure an area of interest [5]. In this SMS, we explored 1852 papers regarding 7 selection criteria, 1831 were discarded and 21 languages were finally selected. These languages are thoroughly studied and classified according to 5 mapping questions and a comparison framework of 12 criteria. Finally we propose some research directions to help developers and researchers in their future works.

Contents. The remainder of this paper is organized as follows. Section 2 shortly introduces the concepts of wireless sensor networks and model driven engineering. Section 3 is devoted for the research mechanism while in Sect. 4, we report the achieved results. Finally, we discuss the principal findings and their implications for researchers and developers before concluding the paper.

2 Background

Wireless Sensor Networks consist of small, cheap, limited battery-powered and spatially distributed sensor nodes which communicate through wireless links. These devices are able of sensing, computing and transmitting real-time information autonomously from the physical area.

Model Driven Engineering is a software engineering approach based on creating and exploiting models in order to address software systems complexity. This approach allows to raise the abstraction level and to discard the low-level details. The model is the core concept in MDE. It is an abstraction of the system under study. The MDE variant proposed by OMG (Object Management Group) is known as MDA (Model Driven Architecture). MDA is based on three models: computation-independent model (CIM), platform-independent model (PIM) and platform-specific model (PSM). MDA transforms a CIM to a PIM and a PIM to a multiple PSMs using Model transformations. A model describes the system using either a general-purpose modeling language such as UML or specific modeling language called Domain Specific Modeling Language (DSML). A DSML is defined using Metamodels or Grammars which underline both the relations between concepts and their static semantics. Thus, the developers can use a DSML to model the system at a higher abstraction level. After that and using Model Transformations, the model can be transformed into another model usually for analysis or code generation purposes.

3 Research Process

To build our research, we decided to conduct a systematic mapping study in order to provide a detailed view of the existing WSNs MDE-based languages. In this study, we follow several steps. The first is the choice of research questions, upon which our future analysis will be carried out. The second step consists in identifying the primary studies that present MDE-based languages for modeling WSNs. Finally, we describe the framework used to compare different languages in the fifth mapping question.

3.1 Mapping Questions

To develop a complete view on the MDE-based languages which are used to model the WSNs area, this study answers five mapping questions. These questions are presented in Table 1, with their corresponding main motivations.

- **MQ1** identifies the publication source and channel for each modeling language.
- **MQ2** aims to classify the languages per year of publication in order to have a clear idea about the research publications trend over time.

Table 1 Mapping questions

	Mapping question	Rationale
MQ1	In which sources and channels are WSNs MDE-based languages published?	To highlight where existing languages can be found
MQ2	What is the publications frequency of WSNs MDE-based languages?	To indicate the publication trends over time of this research area
MQ3	What are the existing MDE-based languages for modeling WSNs?	To identify the existing MDE-based languages that are used to model WSNs
MQ4	What are the main motivations for developing MDE-based languages?	To determine if the language is implemented to facilitate code generation or to allow analysis
MQ5	What are the main characteristics of existing MDE-based modeling languages?	To establish a comparison between the identified languages with respect to a framework of comparison criteria

- **MQ3** provides an overview of the existing MDE-based languages for modeling WSNs. The papers selection criteria are described in the Sect. 3.2.
- **MQ4** delineates the relevant motivation for the language implementation and design. The motivation can be classified as:
  - Code Generation: MDE fosters the design of a platform-independent model that can be used to automatically generate a source code, alleviating the expert developers from the complexity of traditional programming languages.
  - Analysis: MDE allows to transform a system model to another model for analysis purposes (e.g., simulation, model checking, theorem proving).
- **MQ5** establishes a comparison between the different languages. This question allows to identify the features and the weaknesses of each language. So the future studies can be aimed to address these gaps in their languages. The comparison is based on a set of criteria explained in detail in the Sect. 3.3.

3.2 Paper Selection Criteria

To identify the existing MDE-based languages, we explore several search engines and databases, namely, Google scholar, IEEE Digital Library, Science Direct and ACM Digital Library. The following search string is used to conduct the automatic search: ("*Wireless Sensor* Network**" ***OR*** "*Wireless Sensor* and Actuator* Network**" ***OR*** *WSN* ***OR*** WSAN) ***AND*** (("*Model Driven*" ***AND*** (*Engineering* ***OR*** *Architecture* ***OR*** *Development* ***OR*** *Approach**)) ***OR*** "*Domain Specific Language**" ***OR*** "*Domain Specific Modeling Language**" ***OR*** *MDE* ***OR*** *MDD* ***OR*** *MDA* ***OR*** *DSL* ***OR*** *DSML*).

Any paper that satisfies all the inclusion criteria and no one of the exclusion criteria is considered in our investigation.

Inclusion Criteria

- The paper shall be published in English.
- The paper shall be published after 2000 and before 2016.
- The paper's main contribution shall be the introduction to a new MDE-based language for modeling WSNs.
- If a study is conducted on the same modeling language, the latest published paper shall be selected with the aim of staying up-to-date.

Exclusion Criteria

- The paper targets Embedded Systems, Cyber-Physical Systems or Internet of Things (WSN is a part of these systems).
- The paper is built around a modeling language which was previously considered in our investigation.
- The paper involves the state of the art, the summary and the review of existing researches.

3.3 Comparison Framework

In the fifth mapping question, a comparison between the languages is established according to the following criteria:

MDE Instance: What is the MDE instance used in the modeling language? Possible values are MDA or Other. The most used value is MDA.

Modeling Language Origin: The presented language is a new DSML or it is based on a generic existing one. Possible values are DSML or GENERIC.

Modeling Scope: What is the modeling capacity of the selected language? It can model: node level (N) or group of nodes level (G) or network level (Net) or environment (Env). A Group of nodes combines all the nodes that are intended to have the same task in one entity.

Modeling Sensors: Is the language capable of modeling more than one sensor per node? Possible values are Yes or No.

Physical Location: Does the language support the physical location of nodes?

Mobility: Does the modeling language support the mobility of nodes?

Topology: Does the language model the WSNs topology?

Data Aggregation and Fusion: Is the selected language able to model the data aggregation and fusion? Possible values are Yes or No.

Presentation: Does the language model WSNs graphically (GRAPH) or textually (TEXT) or using both of them (MIX)?

Target Language: In case the modeling language is designed to facilitate the code generation, what is the target language?

Evaluation Method: What is the method used to evaluate the language?

4 Research Results

In this section, we report the answers to the mapping questions identified in Table 1. Figure 1 shows the number of produced papers after the execution of the search string in different search engines and databases. Considering the inclusion and exclusion criteria, we select 8 papers from IEEE Digital Library, 6 from ACM Digital Library and 7 from Google Scholar. The 21 selected languages are thoroughly analyzed and discussed.

4.1 MQ1. Publication Sources and Channels

Table 2 showcases the publication sources and channels for the selected MDE-based languages. Most of the sources targeted by selected studies are Conferences and Workshops. They represent 43 % and 29 % respectively. Around 14 % of studies are published in Journals and the same percentage is presented at Symposia. Different channels are identified, but the SESENA Workshop is the unique channel where more than one study are presented. Recall that the SESENA workshop is concerned with topics related to the Software Engineering for Sensor Networks.

4.2 MQ2. Publication Trends

The papers are selected from 2000 to 2016 because WSNs have gained an interest in the twenty first century [1, 3]. Figure 2 shows the number of published languages per year. Generally, we notice that all the languages are published after 2007 and in an increasing way, which can be interpreted by the rise of the third commercial generation of sensor networks technology [6]. This generation is characterized by some properties that attract a plethora of applications, such as small sensor nodes with a long life span, standard wireless protocols and low power-consumption based processors. The demand on such WSNs invites developers to rely on software development, especially the MDE approach, to alleviate their applications design and deployment.

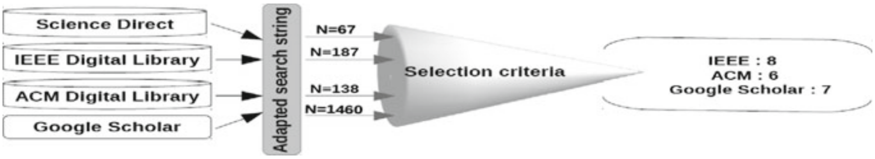


Fig. 1 Selection process

Table 2 MQ1, MQ2, MQ3 and MQ4 results

Language[a]	Publication year	Publication source	Publication channel	Modeling purpose
MEDWSA [10]	2007	Journal	CIS	Code generation
Wada et al. [11]	2007	Conference	SEA	Code generation
ScatterClipse [12]	2008	Symposium	ISPA	Code generation
Escolar et al. [13]	2008	Conference	DCOSS	Code generation
Baobab [14]	2009	Conference	UNISCON	Code generation
Flow [15]	2009	Conference	SENSYS	Code generation
SM4RCD [16]	2010	Workshop	SESENA	Code generation
Moppet [17]	2011	Workshop	BADS	Code generation
Xuan Thang et al. [18]	2011	Conference	MOMM	Code generation
BPMN4WSN [19]	2012	Conference	BPM	Code generation
VeriSensor [7]	2012	Workshop	PNSE	Analysis
Doddapaneni et al. [9]	2012	Workshop	SESENA	Code generation—analysis
Harbouche et al. [20]	2013	Workshop	WETICE	Code generation
SyVad [8]	2013	Symposium	ISPS	Analysis
Lwissy [21]	2013	Workshop	SESENA	Code generation
Vujovic et al. [22]	2014	Symposium	SACI	Code generation
WiSeN [23]	2014	Conference	INDIN	Code generation
Tei et al. [24]	2014	Journal	SMCS	Code generation
WML [25]	2014	Conference	IDCS	Code generation
AMF [26]	2015	Conference	ECMFA	Code generation
ArchWiSeN [27]	2015	Journal	SSM	Code generation

[a] By convention, for unnamed languages, the authors are referenced instead

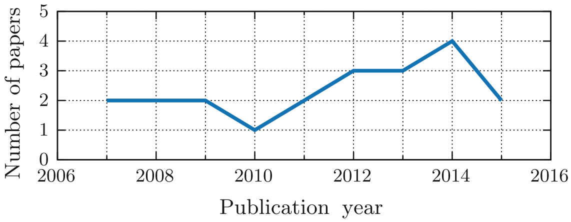


Fig. 2 Number of papers published per year from 2000 to 2016

4.3 MQ3 and MQ4. Modeling Motivations

Table 2 shows 21 selected languages. They are sorted by year of publication for the sake of clarity. Around 90 % of selected languages model the WSNs applications in order to facilitate the code generation. The other percentage represents three languages, VeriSensor [7], SyVad [8] and [9], which are proposed to model applications for analysis purposes. In VeriSensor, Ben Maissa et al. introduce a DSML and its transformation to a formal model (Instantiable Transition Systems) for reliability model checking purposes. In SyVad, Berrani et al. propose a SysML model and its mapping to a Modelica model for simulation and verification purposes, while in [9], Doddapaneni et al. present a DSML and its transformation to Castalia scripts for simulation purposes.

4.4 MQ5. Languages Features

A summary of the main characteristics of the selected languages is presented in Table 3. Around 29 % of selected MDE-based languages adopt the MDA approach, this is a significant value that confirms the idea which presents the MDA as the popular MDE instance. Most languages (62 %) introduce one or more DSMLs to model WSNs applications. This percentage points out that the majority of developers prefer introducing their own languages. As shown in Fig. 3, most of the DSMLs have the option to model WSNs graphically which facilitate the programming task from the cognitive viewpoint. The other 38 % are built on generic languages for designing WSNs. BPMN4WSN [19] extends Business Process Modeling Notation (BPMN) to graphically model WSNs and integrate them with Business Processes. While the 7 other studies propose to add UML stereotypes and profiles. We notice that in the last two years, there is a common trend to use UML.

Around 67 %, 9 % and 5 % of the studies model respectively the node-level, the group-level and the network-level of WSNs applications as shown in Fig. 4. In other studies like [21], several DSMLs are proposed to model the application at different levels which increase the learning cost. The use of group-level or network-level models is more suitable for average developers because they don't require as much details as node-level models. The language which handles the environment aspect is presented by Doddapaneni et al. [9]. They model the physical area where sensor nodes are deployed and present some obstacles that attenuate the transmitting power.

Several papers don't express all the information about their modeling languages. So, no definitive conclusions could be made about the rest of comparison criteria. Around 12 selected languages allow to model several sensors per node. This worthwhile option allows to reduce the number of deployed sensor nodes as well as the hardware cost. Regarding the physical location, it is important to determine the space coordinates of a sensor node. In this way, we can estimate where an event has happened or where an important data is collected. But half of the languages don't model this criterion.

In our mapping study, Harbouche et al. introduce the only language [20] which supports mobility. Indeed, they present a health monitoring system where the collector node was mobile in order to collect the different sensory data from the physiological sensors. Around 11 languages support the modeling of topology whereas 6 languages don't. The WSNs topology allows to understand the nodes positions and their neighborly relations.

Table 3 Languages features

Language[a]	MDE instance	Language origin	MScope	MS	PL	Mob	Top	DA	DF	Presentation	TL	Evaluation method
MEDWSA [10]	MDA	DSML	G	Yes	No	No	Yes	No	No	MIX	NesC	Case study
Wada et al. [11]	Other	GENERIC (UML)	N	No	Yes	No	Yes	No	No	GRAPH	NesC	Case study
ScatterClipse [12]	Other	DSML	N, Net	Yes	No	No	Yes	No	No	MIX	C	Case study
Escolar et al. [13]	MDA	DSML	N	Yes	–	–	–	–	–	GRAPH	NesC	–
Baobab [14]	Other	DSML	N	No	No	No	Yes	Yes	No	TEXT	NesC	Case study
Flow [15]	Other	DSML	N	–	–	–	–	–	–	GRAPH	C	–
SM4RCD [16]	Other	DSML	Net	–	–	–	–	–	–	GRAPH	C++	Experiments
Moppet [17]	Other	DSML	N	No	Yes	No	No	Yes	No	TEXT	NesC	Case study Simulation EE
Xuan Thang et al. [18]	MDA	DSML	N	Yes	No	No	No	Yes	No	MIX	NesC	Case study
BPMN4WSN [19]	Other	GENERIC (BPMN)	N	Yes	No	No	No	Yes	No	GRAPH	–	Case study
VeriSensor [7]	Other	DSML	N	Yes	Yes	No	Yes	No	No	TEXT	–	Case study
Doddapaneni et al. [9]	Other	DSML	N, Env	Yes	Yes	No	Yes	No	No	MIX	–	Case study
Harbouche et al. [20]	Other	GENERIC (UML)	N	–	–	Yes	Yes	Yes	No	GRAPH	NesC	Case study
SyVad [8]	MDA	GENERIC (UML)	N	No	No	No	No	No	No	GRAPH	–	Case Study Simulation

(continued)

Table 3 (continued)

Language[a]	MDE instance	Language origin	MScope	MS	PL	Mob	Top	DA	DF	Presentation	TL	Evaluation method
Lwissy [21]	MDA	DSML	N,G,Net	Yes	No	No	Yes	Yes	No	MIX	NesC	SE
Vujovic et al. [22]	Other	DSML	N	Yes	No	No	No	No	No	MIX	–	–
WiSeN [23]	Other	GENERIC (UML)	N	Yes	Yes	No	Yes	Yes	No	GRAPH	C++	Case study
Tei et al. [24]	Other	DSML	N,G,Net	No	Yes	–	Yes	Yes	Yes	MIX	NesC	Case study User study
WML [25]	Other	GENERIC (UML)	N	Yes	No	No	No	No	No	GRAPH	–	Case study
AMF [26]	Other	GENERIC (UML)	N	–	–	–	–	–	–	GRAPH	NesC	Case study
ArchWiSeN [27]	MDA	GENERIC (UML)	G	Yes	No	No	Yes	Yes	No	GRAPH	NesC	Case study User study

[a] By convention, for unnamed languages, the authors are referenced instead

Acronyms Modeling Scope (MScope), Modeling Sensors (MS), Physical Location (PL), Mobility (Mob), Topology (Top), Data Aggregation (DA), Data Fusion (DF), Target Language (TL), Empirical Experiments (EE), Statistical Experiments (SE)

Fig. 3 Languages origins and presentations

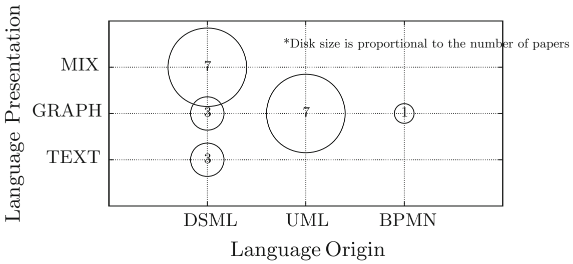
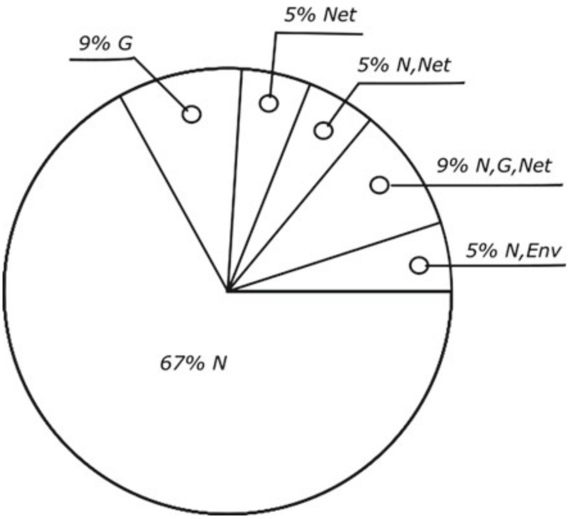


Fig. 4 Languages modeling scope. *Acronyms* Node level (N), Group of nodes level (G), Network level (Net), Environment (Env)



Data aggregation is a process that collects data from multiple sensor nodes, eliminates redundancy using special techniques (e.g., average, maximum) and transmits the reduced data in one copy [28]. Data fusion processes the gathered data in order to achieve relevant, precise and complete inferences [29] (e.g., sensor nodes transmit their own locations and their event detection times which can be fused to obtain the event position). In our study, 9 languages allow to model the data aggregation while just [24] allows the data fusion.

Most languages (58 %) for code generation support nesC as the target code language. nesC is a component-oriented extension to the C language used to design networked embedded systems. All the languages target one language but in [20], Harbouche et al. use two languages, nesC for the sensor nodes and Java for the mobile collector. Around 76 % of the selected studies are validated and evaluated through case studies. In other languages, [24, 27], particular studies are also conducted to test the understandability and the ease of use of the language for the average developers.

5 Discussion

In this section, we analyze the principal findings and the conclusions which can be drawn from the mapping results and present some implications and recommendations for researchers and aspiring languages developers.

5.1 Principal Findings

Analysis. In critical systems where a failure can cause dangerous economic, human or environmental damages, modeling for analysis purposes is very important to realize rigorous and exhaustive verification. However, few studies handle this issue given the following reasons:

1. The exhaustive verification is based on formal models which require a strong experience and skills in mathematics.
2. The process is complicated, because developers must transform a model to another model and conduct analysis. After that, developers must translate the analysis results in the original model language.

Physical Environment. The environment affects strongly the WSNs behavior. But we notice this aspect is considered in only one language [9].

Mobility. The only language which supports the mobility is introduced by Harbouche et al. [20]. In fact, the language models mobile collector which moves throughout the sensing area in order to collect data from sensor nodes. In this context, the mobility allows to reduce the number of transmission hops and thus to minimize the required transmission energy. This result reflects the modeling difficulty of this feature. But, with the growing demand on Mobile WSNs (MWSNs), MDE-based languages must deal with the mobility aspect.

Data Fusion. In energy-constrained sensor networks, sensor nodes can process the data collected from the physical environment. Processing data allows to reduce traffic load and energy consumption and to overcome sensor failures. Note that the data processing requires less energy compared to data communication. Indeed, assuming a 1 GHz carrier frequency, an antenna elevation of 1/2 wavelength, an efficient digital modulation, Rayleigh fading, fourth-power distance loss, 10^{-6} error probability, an ideal receiver and a general-purpose processor with 100 MIPS/W power, transmitting 1kb over a distance of 100 m and executing 3 million instructions consume approximately the same energy [30]. But, according to our study, developers are more attracted to the data aggregation given its modeling ease, than data fusion which is relatively hard to model.

Language Extensibility. In some studies, developers propose instruments to facilitate the language extensibility which encourage other developers to adopt the language and to tailor it to their needs.

5.2 Recommendations for Developers and Researchers

Analysis. Analysis is recommended especially in critical systems. Considering the complicated mathematical aspect of formal verification (e.g., model checking and theorem proving), developers should at least propose more languages for simulation purposes.

Physical Environment. The environment affects strongly the application behavior. For example, the presence of obstacles attenuates the transmitting power, the environment changes (e.g., earthquake, fire) can trigger the nodes mobility or destruction, also the environmental noises as temperature or pressure can affect the data precision. These situations must be considered by developers.

Mobility. Mobility is an important challenge and shall be more explored by developers. Developers can model mobile nodes in several scenarios:

- Sensor nodes which are programmed to move using the mobilizer unit. So, developers can propose a model to the mobilizer unit.
- Sensor nodes which move due to external forces, such as when attached to a vehicle.
- Sinks or collectors which move throughout the sensing region to collect data from sensor nodes. In this case, developers can follow the example proposed by Harbouche et al. in [20].
- Actuator nodes can be mobile to perform actions upon the maximum of the sensing area. These nodes are important in several domains such as environmental monitoring. Note that, in some applications, integrated sensor/actuator nodes are used, which means that the node contains both the sensing and actuating units.

Developers can use some specific mobility models presented by Rezazadeh et al. [31]. These models try to simply present the real behavior of mobile sensor nodes.

Data Fusion. There is no comprehensive theoretical framework to unify the various algorithms proposed in the literature for data fusion [29]. The lack of such unified framework raises the issue of data fusion modeling. The researchers are thus invited to elaborate such framework to alleviate the task of developers. Otherwise, these latter are constrained to model the existing specific algorithms.

Language Unification. Based on our study, we select 21 different MDE-based languages in the WSNs area. Most of them propose instruments to extend the languages to help developers in their future works. However, language unification should receive more attention from researchers to resolve the languages diversity problems and proliferation. This concern has been raised by [32]. Researchers can get inspiration from the UML unification process. UML is an unified modeling language which can be extended by profiles mechanisms.

6 Conclusion

In this paper, we proposed a systematic mapping study on the existing MDE-based languages in the WSNs area. From 1852 surveyed papers, 21 languages were selected according to 7 inclusion and exclusion criteria. These languages were deeply reviewed and analyzed according to 5 research questions and 12 rigorous comparative criteria. Our main results showed that MDE had gained an increasing interest in the WSNs area since 2007 and MDA was the most used MDE instance. The study revealed that 19 languages were designed to facilitate the code generation and just 3 were implemented for analysis purposes. Another interesting result was that nesC was the main targeted programming language. Our study pointed out the lack of some important modeling features such as mobility, environment and data fusion. Finally, we proposed 5 research directions and recommendations to help developers address the languages weaknesses: analysis support, physical environment, mobility, data fusion modeling and language unification.

In the future, along with the existing MDE-based WSNs languages, more studies shall be conducted to provide a better insight into the modeling weaknesses and to propose more solutions according to the proposed recommendations.

References

1. Chong, C., Kumar, S.P.: Sensor networks: evolution, opportunities, and challenges. J. Proc. IEEE **91**, 1247–1256 (2003)
2. Akyildiz, I.F., Can Vuran, M.: Wireless Sensor Networks. Wiley (2010)
3. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor network: a survey. J. Comput. Netw. **38**, 393–422 (2002)
4. Bzivin, J.: On the unification power of models. J. Softw. Syst. Model. **4**, 171–188 (2005)
5. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 68–77 (2008)
6. Sohraby, K., Minoli, D., Znati, T.: Wireless Sensor Networks: Technology, Protocols, and Applications. Wiley-Interscience (2007)
7. Ben Maissa, Y., Kordon, F., Mouline, S., Thierry-Mieg, Y.: Modeling and analyzing wireless sensor networks with VeriSensor. In: International Workshop on Petri Nets and Software Engineering (PNSE), pp. 60–76 (2012)
8. Berrani, S., Hammad, A., Mountassir, H.: Mapping sysml to modelica to validate wireless sensor networks non-functional requirements. In: 11th International Symposium on Programming and Systems (ISPS), pp. 177–186 (2013)
9. Doddapaneni, K., Ever, E., Gemikonakli, O., Malavolta, I., Mostarda, L., Muccini, H.: A model-driven engineering framework for architecting and analysing wireless sensor networks. In: The 3th International Workshop on Software Engineering for Sensor Network Applications (SESENA), pp. 1–7 (2012)
10. Vicente-Chicote, C., Losilla, F., Alvarez, B., Iborra, A., Sanchez, P.: Applying MDE to the development of flexible and reusable wireless sensor networks. J. Coop. Inf. Syst. **16**, 393–412 (2007)
11. Wada, H., Boonma, P., Suzuki, J., Oba, K.: Modeling and executing adaptive sensor network applications with the MATILDA UML virtual machine. In: 11th IASTED International Conference on Software Engineering and Applications, pp. 216–225 (2007)

12. Al Saad, M., Fehr, E., Kamenzky, N., Schiller, J.: ScatterClipse: a model-driven tool-chain for developing, testing, and prototyping wireless sensor networks. In: International Symposium on Parallel and Distributed Processing with Applications, pp. 871–885 (2008)
13. Escolar, S., Carretero, J., Isaila, F., Tartari, G.: A MDA-based development framework for sensor networks applications. In: 4th IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS) (2008)
14. Akbal-Delibas, B., Boonma, P., Suzuki, J.: Extensible and precise modeling for wireless sensor networks. In: 3rd International United Information Systems Conference (UNISCON), pp. 551–562 (2009)
15. Naumowicz, T., Schroter, B., Schiller, J.: Prototyping a software factory for wireless sensor networks. In: 7th ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 369–370 (2009)
16. Glombitza, N., Pfisterer, D., Fischer, S.: Using state machines for a model driven development of web service-based sensor network applications. In: Workshop on Software Engineering for Sensor Network Applications (SESENA), pp. 2–7 (2010)
17. Boonma, P., Suzuki, J.: Model-driven performance engineering for wireless sensor networks with feature modeling and event calculus. In: 3rd Workshop on Biologically inspired algorithms for distributed systems (BADS), pp. 17–24 (2011)
18. Xuan Thang, N., Zapf, M., Geihs, K.: Model driven development for data-centric sensor network applications. In: 9th International Conference on Advances in Mobile Computing and Multimedia (MoMM), pp. 194–197 (2011)
19. Tranquillini, S., Spieß, P., Daniel, F., Karnouskos, S., Casati, F., Oertel, N., Mottola, L., Oppermann, F.J., Picco, G.P., Römer, K., Voigt, T.: Process-based design and integration of wireless sensor network applications. In: 10th International Conference on Business Process Management (BPM), pp. 134–149 (2012)
20. Harbouche, A., Erradi, M., Kobbane, A.: A flexible wireless body sensor network system for health monitoring. In: 22nd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 44–49 (2013)
21. Dantas, P., Rodrigues, T., Batista, T., Delicato, F.C., Pires, P.F., Li, W., Zomaya, A.Y.: LWiSSy: a domain specific language to model wireless sensor and actuators network systems. In: 4th International Workshop on Software Engineering for Sensor Network Applications (SESENA), pp. 7–12 (2013)
22. Vujovic, V., Maksimovic, M., Perisic, B., Milosevic, V.: A Graphical editor for RESTful sensor web networks modeling. In: 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 61–66 (2014)
23. Paulon, A.R., Frohlich, A.A., Becker, L.B., Basso, F.P.: Wireless sensor network UML profile to support model-driven development. In: 12th IEEE International Conference on Industrial Informatics (INDIN), pp. 227–232 (2014)
24. Tei, K., Shimizu, R., Fukazawa, Y., Honiden, S.: Model-driven-development-based stepwise software development process for wireless sensor networks. J. IEEE Trans. Syst. Man Cybern. Syst. **45**, 675–687 (2014)
25. Ruiz-Zafra, A., Noguera, M., Benghazi, K.: Towards a model-driven approach for sensor management in wireless body area networks. In: 7th International Conference on Internet and Distributed Computing Systems (IDCS), pp. 335–347 (2014)
26. Berardinelli, L., Di Marco, A., Pace, S., Pomante, L., Tiberti, W.: Energy consumption analysis and design of energy-aware WSN agents in fUML. In: 11th European Conference on Modelling Foundations and Applications (ECMFA), pp. 1–17 (2015)
27. Rodrigues, T., Delicato, F.C., Batista, T., Pires, P.F., Pirmez, L.: An approach based on the domain perspective to develop WSAN applications. J. Softw. Syst. Model. 1–29 (2015)
28. Maraiya, K., Kant, K., Gupta, N.: Wireless sensor network: a review on data aggregation. J. Sci. Eng. Res. **2** (2011)
29. Abdelgawad, A., Bayoumi, M.: Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks. Springer (2012)

30. Pottie, G.J., Kaiser, W.J.: Wireless integrated network sensors. J. Commun. ACM **43**, 51–58 (2000)
31. Rezazadeh, J., Moradi, M., Ismail, A.S.: Mobile wireless sensor networks overview. J. Comput. Commun. Netw. **2** (2012)
32. Malavolta, I., Muccini, H.: A study on MDE approaches for engineering wireless sensor networks. In: 40th Euromicro Conference on Software Engineering and Advanced Applications (2014)