

A Study on MDE Approaches for Engineering Wireless Sensor Networks

Ivano Malavolta

Gran Sasso Science Institute, L'Aquila, Italy

ivano.malavolta@univaq.it

Henry Muccini

Information Engineering, Computer Science and Mathematics department, University of L'Aquila, Italy

henry.muccini@univaq.it

Abstract—Model-Driven Engineering (MDE) can be considered as the right tool to reduce the complexity of Wireless Sensor Network (WSN) development through its principles of abstraction, separation of concerns, reuse and automation. In this paper we present the results of a systematic mapping study we performed for providing an organized view of existing MDE approaches for designing WSNs.

A total number of 780 studies were analysed; among them, we selected 16 papers as primary studies relevant for review. We setup a comparison framework for these studies, and classified them based on a set of common parameters. The main objective of our research is to give an overview about the state-of-the-art of MDE approaches dedicated to WSN design, and finally, discuss emerging challenges that have to be considered in future MDE approaches for engineering WSNs.

I. INTRODUCTION

Wireless Sensor Networks (WSN) development has been for long time the unique prerogative of embedded systems experts with specialized technical skills, hampering a wider acceptance of this technology. As a way to open WSN development to domain experts, while keeping acceptable levels of efficiency and performance, several programming approaches have been proposed as surveyed in [1]. Still, the *separation of concern* is limited since hardware and software components are locked and tied down to specific types of nodes, hampering the possibility to *reuse* components across projects and organizations. *Abstraction* is also very limited, being the focus on implementing hardware-customized features.

In order to simplify the design and configuration of the WSN at large, and abstract from technical low-level details, a number of Model-Driven Engineering (MDE) approaches for WSN engineering have been proposed. MDE [2] is an effective instrument for expressing domain concepts by abstracting upon selected aspects of the system under consideration and to assess specific properties of the system early in the life cycle. In MDE, domain-specific modeling languages are used to describe the application; they are defined using metamodels, which define both the relations between concepts within the domain and their static semantics. Domain-specific languages are used to build a model of the system that can be successively manipulated in order to perform model assessment, system analysis, or produce various types of artifacts (e.g., source

code, complementary models, deployment configurations, simulation scripts, inputs for analysis tools).

Our goal is to better understand how MDE techniques have been used for designing and analysing wireless sensor networks. In order to elicit and classify existing MDE approaches, this work surveys and classifies state-of-the-art MDE approaches for engineering a WSN. Through a systematic mapping study, we select 16 primary studies (among 780 entries) fitting at best two research questions we identified. Then, we define a classification framework comprehending 15 different parameters for comparing state-of-the-art approaches. Finally, we discuss emergent challenges that have to be considered in future MDE approaches for engineering WSNs.

Basically, the main challenges we identified from our research are:

- many approaches propose their own ad-hoc modeling language for representing a WSN, researchers should avoid this proliferation of different modeling languages in favor of an (extensible) standard language for WSNs;
- almost all studied approaches are built on a single monolithic modeling language comprising all the concepts to model the WSN, researchers should focus on a better separation of concerns when dealing with WSNs;
- almost all the presented approaches do not provide means for modeling nodes mobility, researchers should support this increasingly relevant aspect of WSNs;
- many approaches mix together notions and concepts coming from both MDE and WSN communities, making those approaches very hard to understand from members of both communities, MDE researchers should take care in masking the complexity of the used MDE techniques to WSN engineers.

In light of those challenges, this paper can certainly be useful for researchers and practitioners contemplating to use MDE in the construction of WSN systems. Indeed, MDE seems to be quite practical and useful for the construction of WSN systems, however the research at the cross-section of MDE and WSE has been rather sparse. This paper aims to shed some light on the existing research and challenges for future research.

The main contributions of this study can be summarized as in the following:

- definition of a comparison framework for past and future MDE approaches for WSNs;
- a systematic overview of current MDE approaches for engineering WSNs, useful for both researchers and practitioners;
- a discussion of emerging research challenges for future MDE approaches for WSNs.

The paper is structured as follows: Section III identifies the main research questions. The research instrument and the comparison framework are described in Section IV. The results are presented and discussed in Section V, while the scope and limitations of the study are presented in Section VI. Future research challenges for WSNs are drawn in Section VII. Final remarks and conclusions are presented in Section VIII.

II. BACKGROUND

Wireless Sensor Networks consist of spatially distributed sensors that cooperate to accomplish some tasks. Sensors are small battery-powered devices with limited processing capabilities and memory. In the current state of the art, WSN nodes processor frequencies range from 4Mhz to 32 Mhz, whereas typical amounts of memory range from 2 Kb to 512 Kb [1]. They can be easily deployed to monitor different environmental parameters such as temperature, movement, sound and pollution. Sensors can be distributed on roads, vehicles, hospitals, buildings, people and enable different applications such as medical services, battlefield operations, crisis response, disaster relief and environmental monitoring.

The unique characteristics of WSNs introduce new challenges [3] in different fields such as programming, security and software engineering. Not only researchers need to face limited sensor resources in terms of computation capabilities and memory but they also need to face the limited life time of the sensors.

Model-Driven Engineering (MDE) approaches for Software Architecture descriptions have been proposed. MDE [4] is an effective instrument for expressing domain concepts by abstracting upon selected aspects of the system under consideration and to assess specific properties of the system early in the life cycle. In MDE, domain-specific modeling languages are used to describe the application; they are defined using metamodels, which define both the relations between concepts within the domain and their static semantics. Domain-specific languages are used to build a model of the system that can be successively manipulated in order to perform model assessment, system analysis, or produce various types of artifacts (e.g., source code, complementary models, deployment configurations, simulation scripts, inputs for analysis tools).

III. RESEARCH QUESTIONS

We designed our investigation with two research questions in mind:

- 1) *RQ1*. What are the existing MDE approaches for modeling, analysing and developing WSNs?

- 2) *RQ2*. How do existing MDE approaches for modeling analysing and developing WSNs compare?

RQ1 concerns the identification and selection of current MDE approaches for WSNs with a special focus on those approaches that (i) are based on a modeling language for WSN design and (ii) perform either analysis, code generation, or estimation starting from the WSN design models. The selection criteria are described in Section IV.

RQ2 explores how the previously selected MDE approaches compare with respect to a set of reference parameters. Examples of parameters include: supported semantic analyses, supported middleware, whether the approach supports mobile nodes, whether the approach allows designers to model the physical environment, etc. We defined those parameters as a comparison framework, which will serve as the basis for analysing and comparing the selected MDE approaches in a rigorous manner. The comparison framework is described in Section IV.

To answer these research questions, we decided to conduct a systematic mapping study. This kind of research is extremely powerful in collecting and analysing existing work since it helps in selecting and summarizing the existing evidence, while minimizing as much as possible the chances of bias [5].

To the best of our knowledge, this paper presents the first investigation into the usage of Model-Driven Engineering for modeling, analysing, and developing wireless sensor networks.

In the literature, there are some other works which in principle could be seen as similar to ours, however they mostly differ in either the main research questions, in the subjects of the research or in how the research itself has been carried on. More specifically, in [1], the authors survey the state of the art in how WSNs can be programmed, rather than focussing on how to model WSNs for analysis, code generation etc. Examples of considered aspects in that survey include: communication and computation scope, communication addressing, data model, programming paradigm (e.g., imperative, declarative, hybrid).

Also in [6] the focus is on approaches to programming wireless sensor networks, rather than on how to model them. In that work the surveyed approaches are firstly classified into low-level and high-level ones, and then they have been analysed with respect to recurrent requirements for WSNs, such as energy-efficiency, scalability, failure-resilience.

Finally, in [7] a survey about modeling techniques for WSNs is also presented; there, the authors classify nine approaches with respect to properties such as the used notation, the modeling scope, whether the approach allows designers to model the topology of the network etc. Differently from [7], our investigation is specifically tailored to Model-Driven Engineering approaches; also, our work is more rigorous since we applied the guidelines suggested in [5].

IV. THE RESEARCH INSTRUMENT

In this section we describe how we selected the subjects of our research and present the comparison framework we use for classifying the identified approaches.

A. Approaches Selection

The first step of our investigation consists in selecting those research articles presenting a model-driven engineering approach for designing a wireless sensor network. To achieve this, we defined a set of *Google Scholar*¹ queries to obtain an initial list of potential candidate articles. We use Google Scholar since it enabled us to thoroughly search peer-reviewed papers, theses, books, preprints, abstracts and technical reports. Google Scholar has been particularly useful because it allowed us to be reasonably confident about the completeness of the results, collecting them from the web, academic publishers (such as ACM, Springer, IEEE), professional societies repositories, and universities. Also, it is important to note that Google Scholar performs a full-text search (as opposed to document title and summaries) when performing our search queries, this gives us the guarantee that the obtained research works are relevant with respect to the performed queries and that their whole contents is related to some degree to our object of study (i.e., MDE approaches for WSNs).

In order to achieve a reasonable coverage of research works related to MDE approaches for WSNs we performed four different queries on Google Scholar; we present them in Listing 1. When going through the results of those queries, we considered only those documents which have been published from 2007 to the present day since we wanted both to have a manageable set of results, and to focus on state-of-the-art approaches only.

```
1 wsn AND mde
2 mde AND wireless AND sensor AND network
3 model AND driven AND engineering AND wsn
4 model AND driven AND engineering AND wireless AND sensor
  →AND network
```

Listing 1. Performed Queries on Google Scholar

Basically, we firstly searched for articles containing **both** the terms *wsn* and *mde* (see line 1 of the listing) we crafted the query in that manner because in this research we are focussing on the use of MDE approaches in the WSN domain, rather than on articles focussing on MDE or WSN separately. This query performed on the *Engineering, Computer Science, and Mathematics* domain returned a list of 130 references. Then, we performed all the other queries reported in lines 2, 3, and 4 of Listing 1 in order to include also those articles containing long versions of the WSN and MDE acronyms. Those queries produced 403, 83, and 164 hits, respectively.

The combined output of these queries is a collection of 780 (possibly repeated) documents. Then, we merged all the documents with the same title and authors into a single one. After this step, we read and analysed all the titles and abstracts of the obtained references in order to identify the primary studies relevant for review. In order to identify those primary studies that provide direct evidence about our research questions, in this phase we considered the following inclusion and exclusion criteria:

• Inclusion criteria.

- Any article declaring that its main contribution is the definition of a new MDE approach for WSNs.
- Any article that have been published in or after 2007.
- Any article that have been published in English.
- **Exclusion criteria.**
 - Articles that have been extended by another article that have been previously considered in our survey. This allows us to have a unique representative article for each approach.
 - Articles that do not present any specific approach in details. In this category lie articles that discuss the use of MDE for modelling WSNs in general, presenting a survey or a comparison between other approaches
 - Articles with incomplete information about our comparison parameters presented in Section IV-B.
 - Articles that are an editorial, abstract, position paper, short paper, tool paper, poster summary, keynote, opinion, tutorial, introduction to conference proceedings, workshop summary, panel summary. All these articles do not provide a reasonable amount of information.
 - Articles that are not validated by the Computer Science community, such as technical reports and white papers.

In summary, our primary studies must present an MDE approach for either modeling, designing, analysing or developing WSNs with a strong focus on the models of the WSN (i.e., an approach which is *driven* by the model of the WSN). After applying the selection criteria, 16 articles were identified as primary studies².

B. The Comparison Framework

In our research we analysed the WSN domain with a focus on programming abstractions and modeling, and then identified a set of parameters for benchmarking the selected MDE approaches with respect to research question *RQ2*. The set of parameters initially identified in our comparison framework is orthogonal to the scope and applicability of each MDE approach for WSN. Indeed, regardless of the focus of each specific MDE approach, in general its main characteristics, such as the presence of a dedicated modelling language, the support of mobility of nodes, or the kinds of analysis that can be performed on its models, can be considered as a constant. As might be expected, a large number of parameters came out, each one focusing on specific aspects of an MDE approach for WSN, its implied activities, and methodologies. In order to organize those parameters, we classify them into three clusters, built by considering the three main viewpoints from which an MDE approach can be analysed [8]:

- 1) *Modelling language features*: this cluster focusses on the main characteristics of the used modelling language concerning its expressive power, the viewpoint from which it frames the WSN being modelled, and how it

¹Google Scholar website: <http://scholar.google.com>.

²A summary of the selected articles is available here: <http://goo.gl/eCw2>

abstracts from the details of the implementation of the WSN.

- 2) *Goals*: this cluster focusses on the main motivation why a practitioner of researcher should used the MDE approach; mainly, it classifies the MDE approach with respect to the supported activities in the development life cycle (e.g., code generation, documentation, etc.), the type of analysis it supports (e.g., performance analysis, security analysis, etc.), and the target implementation code if the MDE approach supports code generation.
- 3) *Technological aspects*: this cluster focusses on those aspects of the MDE approach which are more related to the implementation of the approach, such as its modelling environment, the presence of a tool, the presence of a graphical editor, and so on.

In the following we further elaborate on each cluster of the comparison framework.

Modeling Language Features

Modeling Language. Does the approach propose a new dedicated language for WSNs (*DSML*), or is it built on a generic existing one (*GENERIC*), for example UML?

Structure & Behaviour. Is the MDE approach capable of modeling the structure or the behaviour of the designed WSN?

Mobility. Does the approach deal with WSN nodes mobility? By taking inspiration from [1], we identified three possible values: static (*S*), mobile nodes (*MN*), mobile sinks (*MS*).

Computation Scope. How does the approach represent the computation within the WSN? We distinguish between node-level (*N*), group-level (*G*), or network-level (*NET*) computation scopes [1].

Abstraction level. By focussing on the reference architecture provided in [1, Section 3], here keep track of the lowest level of abstraction that can be represented by the modeling language of each MDE approach. Possible values of this parameter include: hardware (*H*) MAC, operating system (*OS*), system service (*S*), and application (*A*).

Physical Deployment. Does the MDE approach allow designers to specify the physical deployment of the WSN nodes? Possible values are true or false.

Power Consumption. Does the MDE approach allow designers to represent the properties related to power consumption, energy draining, etc.? Possible values are true or false.

Location Awareness. Are the WSN nodes aware of their position within the network? Possible values are true or false.

Goals

Goals. What are the main goals of the MDE approach? Values are defined within this set: code generation (*CO*), analysis (*AN*), test cases generation (*T*), documentation (*D*).

Analysis Type. If the goal of the approach is analysis, which kind of analysis are supported? Values of this parameter include: performance (*PE*), fault tolerance (*FT*), power consumption (*PO*), security (*SEC*).

Target Language. If the goal of the approach is code generation, which target languages are supported? Here possible values are: C++, NesC, etc.

Technological Aspects

Used Technologies. What are the technologies used to implement the MDE approach? Examples of possible values include: Eclipse, standalone application, etc.

Concrete Syntax. Are the models represented graphically (*GRAPH*), textually (*TEXT*), or a combination (*MIX*) of them?

Extensibility. Is the MDE approach extensible in some way? If so, is the exposed modeling language extensible (*L*), or is the underlying framework extensible (*F*)?

V. MDE APPROACHES FOR WIRELESS SENSOR NETWORKS

We examined each primary study by clustering them through those parameters we identified in Section IV-B.

A. Modeling Languages Features

Table I shows the main features of the modeling languages used in each MDE approach. Around 75% of the selected approaches (12 out of 16) propose to use **Domain-specific Modeling Languages** (*DSML*) for representing WSNs from different viewpoints. For example, in [9] the proposed language contains concepts such as node group, region, resource, wireless link; whereas, in [10] authors propose a set of languages spanning from application-level actions (e.g., sense, send message, store data) to hardware specifications (e.g., processor, sensing devices, radio transceivers), and so on. The remaining 25% are based on *generic modeling languages*; mainly, they use extensions of UML and Simulink for representing a WSN.

We can notice a certain balance when considering the **structure** (9 approaches out of 16) and **behaviour** (12 approaches out of 16) parameters. This result makes sense, since describing a network from a structural point of view is very straightforward and easy to reason on (just think about the component-based representation in Omnet++³, one of the most used network simulators). Also, in some cases (e.g., when capabilities such as fault tolerance and security analysis are needed) the structure of the WSN may be not enough, and thus describing the behaviour of the WSN is fundamental.

For what concerns **mobility**, there are no approaches which allow designers to specify mobile nodes (and actually only one approach, [11], supports mobile sinks). This is quite surprising, especially in light of the great momentum these kind of WSN applications are getting in the last years.

Our analysis about the **computation scope** of the selected approaches shows that there is a certain prevalence (8 out of 16) of approaches with a node-level computation scope, followed by the approaches with group-level computation (5 out of 16) and those with a network-level computation (2 out of 16). Interestingly, the modeling approach proposed in [12] encompasses all the three computation scopes, each of them supported by a dedicated modeling language; in that approach, designers create models with node-level computation scope, while models with group- and network-level computation scopes are automatically generated.

³<http://www.omnetpp.org/>

Selected approaches provide languages with varying **abstraction levels**. The majority of those approaches (6 out of 16) allow designers to reason at all levels, down to the hardware of the WSN nodes. Other approaches support to model WSN applications down to the system service (*S*, 5 out of 16), MAC (3 out of 16), and operating system (*OS*, 1 out of 16). This result is quite reasonable since the strict requirements of WSNs, especially on energy consumption [13], ask designers to reason at different levels of abstraction. In light of this, it is quite surprising to see an approach, namely A2, that focusses on the application layer only. A2 is based on BPMN, and it is different from other approaches since its goal is not on the classical notion of design, rather it aims at providing a conceptual model that abstracts typical WSN programming knowledge into reusable tasks that can be integrated into modeling notations [14].

The majority of selected approaches (9 out of 16) does not allow designers to specify the **physical deployment** of the WSN nodes, whereas six approaches out of sixteen support this feature in some way. Among those approaches there is great variability. There are some which support an explicit definition of the physical environment (e.g., A3 [10] with its real-world dimensions, obstacles with attenuation coefficients, etc.), others that allow designers to define physical quantities (e.g., A1 [11] with its models of the evolution of each physical quantity in a given scenario), and so on.

Power consumption can be considered as one of the main challenges in the WSN research domain. This is confirmed also by the number of recent research events dedicated to this theme, like the first international workshop on engineering energy efficient WSNs (EEEWSN⁴) or the third international workshop on green and sustainable software (GREENS⁵). As expected, there is a certain number of approaches (6 out of 16) supporting the definition of properties related to power consumption (such as, sensor readings energy consumption, battery levels, antenna transmission power, energy-related operational modes). However, the majority of the selected approaches (9 out of 16) does not deal with power consumption; all of them compensate somehow this gap by providing code generation facilities, and thus relying on more experimental development settings based on test beds, easy-to-build prototypes, and so on.

WSN node **location awareness** is one of the basic properties for designing smarter WSN applications. Indeed, they allow designers to create WSNs in which a node can behave differently based on its current location within the network. The network itself may reconfigure autonomously depending on the current location of its nodes, and so on. Unfortunately, the majority of our selected approaches (12 out of 16) does not support this feature. Among those approaches supporting nodes location awareness, A7 stands out. A7 allows to model

GPS-enabled nodes that can perform actions (such as data aggregation) depending on the actual geographical position of the node.

B. Main Goals

Table II shows the results of our analysis about the **main goals** of selected MDE approaches; there, we see many interesting trends. For example, performance (*PE* in Table II) and power consumption (*PO* Table II) analyses are very recurrent (10 and 5 approaches out of 13, respectively), as well as nesC code generation (8 approaches out of 13). These trends make sense since performance and power consumption are among the most intriguing challenges in WSN research [3]; also, nesC is the clear winner among code generation target languages since it is the *de facto* standard for developing WSN applications [1].

Moreover, there are many approaches providing analysis (AN in Table II) and code generation (CO in Table II) facilities at the same time (11 approaches out of 16), which is a good indication that modeling languages which are generic enough to support both analysis and code generation exist in MDE.

Approach	Goals	Analysis Type	Target Language
A1 [11]	AN,D	PE,PO,SEC	-
A2 [14]	CO	-	nesC
A3 [10]	AN,CO	PO	-
A4 [15]	AN,CO	PO	nesC,Java
A5 [16]	CO	-	nesC
A6 [12]	AN,CO	PE	nesC
A7 [17]	AN,CO	PE,PO	nesC
A8 [18]	AN,CO,T	PE	nesC
A9 [19]	AN,CO	PE	C
A10 [20]	AN,CO,T,D	PE	-
A11 [21]	AN,D	PE	-
A12 [22]	AN,CO	PE,SEC	C++
A13 [23]	AN,T	FT	-
A14 [24]	AN,CO	PE,PO	nesC
A15 [25]	AN,CO	PE	C
A16 [9]	CO	-	nesC

TABLE II
GOALS

It is also interesting to note that there are no modeling approaches for documentation purposes only (D in Table II); this is in line with the fundamental model-driven engineering principle saying that: “*models and MDE tools serve as a unifying vehicle to document, analyze, and transform information systematically at many phases throughout a system’s life cycle*” [2]. It is clearly in contrast with the (wrong) perception that modeling can be used exclusively for documentation and communication purposes.

C. Technological Aspects

From a technological point of view, the overall picture is quite fragmented. Table III shows the results of our analysis.

⁴<https://sites.google.com/site/eeewsn>

⁵<http://greens.cs.vu.nl>

Approach	Modeling Language	Structure	Behaviour	Mobility	Comput. Scope	Abstr. Level	Physical Dep.	Power Cons.	Loc. Awareness
A1 [11]	DSML	✓	✓	MS	N	H	✓	✓	✓
A2 [14]	GENERIC	✗	✓	S	N	A	✗	✓	✗
A3 [10]	DSML	✓	✓	S	N	H	✓	✓	✗
A4 [15]	DSML	✗	✓	S	G	H	✓	✗	✗
A5 [16]	DSML	✓	✗	S	N	S	✗	✗	✗
A6 [12]	DSML	✓	✓	S	N,G,NET	MAC	✓	✗	✓
A7 [17]	DSML	✗	✓	-	N	H	✗	✓	✓
A8 [18]	DSML	✓	✗	S	-	H	✗	✗	✗
A9 [19]	GENERIC	✗	✓	S	G	S	✗	✗	✗
A10 [20]	DSML	✓	✗	-	N	H	✗	✓	✗
A11 [21]	GENERIC	✗	✓	S	-	MAC	-	-	-
A12 [22]	DSML	✗	✓	-	NET	MAC	✗	✗	✗
A13 [23]	DSML	✓	✓	S	N	S	✗	✗	✗
A14 [24]	DSML	✓	✗	S	G	OS	✓	✓	✗
A15 [25]	GENERIC	✗	✓	S	L	S	✓	✗	✗
A16 [9]	DSML	✓	✓	S	G	S	✓	✗	✗

TABLE I
MODELING LANGUAGES FEATURES

More specifically, 8 out of 16 approaches have been **implemented** via the Eclipse⁶ platform, an open-source development platform comprised of extensible frameworks and tools for building, deploying and managing software across the life cycle. For what concerns the other eight approaches, we did not manage to get any clue about the technologies used to implement them.

Approach	Used Technologies	Concrete Syntax	Extensibility
A1 [11]	-	TEXT	NO
A2 [14]	-	GRAPH	-
A3 [10]	Eclipse	MIX	F
A4 [15]	Eclipse	GRAPH	F
A5 [16]	Eclipse	MIX	L
A6 [12]	-	TEXT	-
A7 [17]	Eclipse	TEXT	-
A8 [18]	-	GRAPH	-
A9 [19]	-	MIX	F
A10 [20]	Eclipse	MIX	F
A11 [21]	-	GRAPH	-
A12 [22]	-	GRAPH	-
A13 [23]	Eclipse	MIX	L
A14 [24]	Eclipse	TEXT	L
A15 [25]	-	MIX	F
A16 [9]	Eclipse	GRAPH	NO

TABLE III
TECHNOLOGICAL ASPECTS

As every computer scientist knows, a (modeling) language is composed of (i) *abstract syntax*, i.e., the set of language concepts and their relationships, (ii) a set of *concrete syntaxes*, i.e., the notations to visualize and edit the models, and (iii) *semantics* describing the meaning of the languages' constructs. In our study we do not compare MDE approaches by their ab-

stract syntax and semantics because they are somehow hidden to end users, whereas we are able to directly compare their exposed **concrete syntaxes**. Fundamentally, we see a balance between textual syntax, graphical syntax, and a combination of them. However, for what concerns the graphical syntax we see some variability. For example, A8 provides a graphical notation similar to state machines, in A11 authors reuse UML activity diagrams, A16 presents a notation which is close to a component-based flow diagram with custom icons, etc.

When considering modeling approaches which are tailored to solve a specific problem like that of WSN design, the provided modeling language may not cover all the possible concerns. One of the classical solutions to this problem is to provide some level of **extensibility**, so that designers can extend either the modeling language or its underlying framework in order to fulfil their needs. In our study, many papers do not give enough information about it, making very difficult to draw some conclusions about this important aspect of a WSN modeling approach. However, in some approaches such as A9 and A3, authors clearly state that their proposed modeling framework is extensible. Other approaches like A5, A13, and A14, propose a modeling approach based on an extensible language.

In conclusion, if on one side, the Eclipse platform is the most used platform for developing the tools supporting the WSN modeling activity, on the other side we see a lot of fragmentation when considering both the concrete syntax of the modeling languages and their capability of being extended.

VI. RESEARCH VALIDITY AND LIMITATIONS

In this section we describe the validity threats to the design of this study.

Data sources. Our choice of data source is the Google Scholar platform. Even though it gives us some benefits (such as being able to query a single data source that is capable

⁶<http://www.eclipse.org>

of automatically fetching results from a number of different databases, academic publishers, and other technical sources, etc.), we are aware that this choice may be a threat to validity to the design of our study since Google Scholar suffers from some known limitations [26] like, among the others, the partial control on the contents of the body of knowledge, the lack of an advanced search facility, its vulnerability to spam. In light of this, when extending this work we will include more data sources, publishers, and science-specific search engines. Likely, this extension will result in more resulting data, thus enabling a more detailed statistical analysis and a broader coverage of the publications about MDE approaches for WSNs.

Data collection process. When collecting and analysing the articles resulting from our initial queries we mostly used a single researcher to review each article (more specifically, its title, abstract, and venue). We tried to alleviate this potential threat to the validity of our study by taking notes of unclear issues and discuss them together about how to solve them individually. We are aware that there still is the risk that a single reviewer can be biased and thus injecting some inconsistencies in the extracted information. As future work, we are planning to perform a more detailed statistical analysis and to have at least more than one reviewer for each article.

Comparison parameters selection. As described in Section IV-B, we identified a set of comparison parameters for classifying MDE approaches for WSN. The selection of those parameters comes from a preliminary analysis we performed of the WSN domain with a focus on programming abstractions and modeling. Even though we performed an accurate analysis, we are aware that there could be also other important features that need to be considered. In order to alleviate this potential threat, during the analysis of the primary studies the researcher was in charge of keeping track of possible recurrent patterns and features of the approaches in the primary studies that could motivate the inclusion of additional parameters in the comparison framework. At the end of the analysis phase, no additional parameter have been found, thus making us reasonably confident about the completeness of the initial comparison framework.

Time span In Section IV-A we highlighted that we considered only those articles which have been published from 2007 to the present day. If on the one side this choice allows us to both have a reasonable number of results and to focus only on the most recent approaches, on the other side it may affect the completeness of our search results since our study does not consider research published before the year 2007. As future work, and hoping to have enough resources for employing more researchers on this study, we are planning to extend this study in order to consider articles published in a wider range of years.

Inference of missing information. During the analysis of the articles we found that some papers did not provide enough explicit information to properly collect the data as described in the comparison framework presented in Section IV-B. In order to mitigate this potential issue, we had to infer certain

information during the analysis of the articles by referring to other articles describing the same approach presented in the analysed one (we matched the approaches by their name), or by accessing the website of the presented approach (if any), or by directly contacting the authors of the articles. Finally, each decision about the inference of missing information has been taken via a discussion among the researchers with the aim to clarify potential ambiguities or misunderstandings.

VII. FUTURE CHALLENGES FOR ENGINEERING WSNs

This research allows us to infer and propose the challenges that have to be considered in future MDE approaches for engineering WSNs.

Firstly, we notice that many approaches propose their own modeling language for representing a WSN. This is mainly due to their wide diversity in terms of supported analyses (like performance, security, etc.), abstraction level (e.g., node-level, group-level), and goal (like code generation, analysis, test case generation). The lack of a satisfactory **standard language for modeling WSNs** can be another reason. What is important to avoid is the proliferation of different modeling languages. If a generic-enough standard language for WSNs could be defined, the problem of WSN languages diversity would be mitigated, thus cutting training costs, lowering the learning curve, unifying development processes, and so on. Such a generic language will however require some extension mechanisms (as those presented in [27] for the software architecture description domain) enabling the language customization to specific stakeholders' concerns.

Secondly, we notice that almost all studied approaches are built on a single monolithic modeling language comprising all the concepts to model the WSN. This is clearly in contrast with the always-desired property of **separation of concerns**, which promotes the modeling and reasoning of a given problem from a set of given viewpoints, each of them focussing on some specific aspect of the system under consideration. In this respect, we suggest WSN and MDE researchers to work together in order to provide future approaches with greater care in cutting the WSN engineering problem in to views and viewpoints, and thus making the development of WSNs more systematic and disciplined.

Thirdly, it is quite relevant to notice that almost all the presented approaches do not provide means for modeling nodes **mobility**. This can be considered an important missing modeling feature, especially in the context of the growing mobile sense and computing paradigm. We are aware that developing a modeling approach supporting mobility is intrinsically hard and that there is little research literature about modeling mobile entities even outside the WSN research domain; however, future MDE approaches should provide this feature in order to support the design of WSN applications with mobile nodes.

Finally, we also noticed that many surveyed approaches seldom try to characterize their final users distinguishing between MDE and WSN experts. As a result, many approaches mix together (usually advanced) concepts coming from both

MDE and WSN communities. This makes those approaches very hard to understand from members of both communities. Thus, **masking the complexity** of the used MDE techniques to WSN engineers is a must for future MDE approaches for WSNs.

VIII. DISCUSSION AND CONCLUSIONS

This study has compared state-of-the art MDE approaches for engineering WSNs, and identified four main challenges to be managed in order to enable a better symbiosis between MDE techniques and WSN engineering.

What appears evident from this study is that some relevant effort is being put by researchers on the use of MDE approaches for the design, analysis, and development of WSNs. Many MDE approaches with different goals and characteristics have been proposed, each devoted to look at the WSN domain from a specific angle. Still, many challenges need to be faced. So, *how to proceed?*

First, our long term experience in other domains (such as the software architecture, and verification and validation ones) suggests that a research community shall be created around the MDE for WSN engineering topic, so to bring together the different perspectives coming from different researchers in the field. A proliferation of (non-standard) notations and tools has been experienced by the software architecture community (see e.g. the study conducted in [28]) and this lead to a limited impact of architectural languages into the industrial practices. Differently, the standardization of the UML for the design of OO systems has had a relevant impact onto industrial practices. In this respect, what the authors take the liberty to propose is a working meeting among the proponents of MDE techniques for the design and analysis of WSN. Practitioners and notes producers shall also be invited into such a meeting, so to design an extensible modeling framework for the engineering of WSNs.

Second, another limitation that has been pointed out consists in the limited separation of concerns between the proposed modeling languages, and limited care on the different WSN stakeholders specific concerns. In this line, the authors would suggest the community interested into this topic to base their current or future work on the ISO/IEC/IEEE 42010:2011, *Systems and software engineering — Architecture description* [29] standard. The ISO/IEC/IEEE 42010 is the joint ISO and IEEE revision of IEEE Std 1471, first published in 2000. It addresses the definition of new multi-view languages to meet the diverse needs of different system's stakeholders. The standard requires to take stakeholders' needs and concerns into account when defining a modeling framework. A multi-view approach is then seen as the natural output, where each view is realized to satisfy concerns of specific stakeholders.

Such a multi-view approach can be implemented through different means: a domain-specific modeling language (DSML) for WSN, or a specialization of a generic language such as the UML. Both bring pros and cons, such as the better fit of a DSML with respect to the WSN modeling needs,

or the re-usability of UML skills and knowledge. Instead of proposing our own (biased) perspective, we would like such an option to be discussed inside the research community to be formed around this topic. In this context, it is important to note that in both cases the problem of identifying, predicting and evaluating the significance of a language change impact over the existing artifacts (e.g., other views, model transformations, analysis tools) still exists [30].

Third, in the era of the internet of things, support for mobility seems to be still very limited. Future work shall have to contemplate two mobility types: *notes mobility* and *code mobility*. For the first case, new models have to be designed, enabling the run-time specification of the changing environment, and how such a changes impact the overall WSN qualities. For the latter case, code mobility in wireless notes, coordination models and code mobility patterns (such as those presented in [31]) could be adjusted to the WSN domain.

Concluding, an MDE approach for engineering WSN have to be understandable, usable, and have to bring knowledge to the different stakeholders involved in the development of a WSN-based application. As a precondition to this, the communities involved in engineering WSNs have to avoid the (as-usual) proliferation of different notations and prototypal tools, while concentrating on a standard (still, extensible) modeling languages.

ACKNOWLEDGMENTS

This work is partly supported by the Ricostruire project (RIDITT - Rete Italiana per la Diffusione dell'Innovazione e il Trasferimento Tecnologico alle imprese). The authors would like to thank Dario Mazzelli, for his help and precious support.

REFERENCES

- [1] L. Mottola and G. P. Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Comput. Surv.*, vol. 43, pp. 19:1–19:51, Apr. 2011.
- [2] D. C. Schmidt, "Guest editor's introduction: Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25–31, Feb. 2006. [Online]. Available: <http://dx.doi.org/10.1109/MC.2006.58>
- [3] J. A. Stankovic, "Research challenges for wireless sensor networks," *SIGBED Rev.*, vol. 1, no. 2, pp. 9–12, Jul. 2004. [Online]. Available: <http://doi.acm.org/10.1145/1121776.1121780>
- [4] S. Kent, "Model driven engineering," in *Proceedings of the Third International Conference on Integrated Formal Methods*, ser. IFM '02. London, UK, UK: Springer-Verlag, 2002, pp. 286–298. [Online]. Available: <http://dl.acm.org/citation.cfm?id=647983.743552>
- [5] "Guidelines for performing systematic literature reviews in software engineering," 2007.
- [6] R. Sugihara and R. K. Gupta, "Programming models for sensor networks: A survey," *ACM Trans. Sen. Netw.*, vol. 4, no. 2, pp. 8:1–8:29, Apr. 2008. [Online]. Available: <http://doi.acm.org/10.1145/1340771.1340774>
- [7] J. K. Jacob, R. Liscano, and J. S. Bradbury, "A survey of modeling techniques for wireless sensor networks," in *Proc. of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM 2011)*, Aug. 2011, pp. 103–109.
- [8] D. Harel and B. Rumpe, "Meaningful modeling: What's the semantics of "semantics"?" *Computer*, vol. 37, no. 10, pp. 64–72, Oct. 2004. [Online]. Available: <http://dx.doi.org/10.1109/MC.2004.172>
- [9] C. Vicente-Chicote, F. Losilla, B. Álvarez, A. Iborra, and P. Sánchez, "Applying MDE to the Development of Flexible and Reusable Wireless Sensor Networks," *Int. J. Cooperative Inf. Syst.*, vol. 16, no. 3/4, pp. 393–412, 2007.

- [10] K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostarda, and H. Muccini, "A model-driven engineering framework for architecting and analysing Wireless Sensor Networks," in *SESENA*, 2012, pp. 1–7.
- [11] Y. Ben Maïssa, F. Kordon, S. Mouline, and Y. Thierry-Mieg, "Modeling and Analyzing Wireless Sensor Networks with VeriSensor," in *Petri Net and Software Engineering (PNSE)*, vol. 851. Hamburg, Germany: CEUR, Jun. 2012, pp. 60–76.
- [12] R. Shimizu, K. Tei, Y. Fukazawa, and S. Honiden, "Model driven development for rapid prototyping and optimization of wireless sensor network applications," in *Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications*, ser. SESENA '11. New York, NY, USA: ACM, 2011, pp. 31–36. [Online]. Available: <http://doi.acm.org/10.1145/1988051.1988058>
- [13] G. P. Picco, "Software engineering and wireless sensor networks: happy marriage or consensual divorce?" in *Proc. of the FSE/SDP workshop on Future of software engineering research*, ser. FoSER, NY, USA, 2010.
- [14] S. Tranquillini, P. Spieß, F. Daniel, S. Karnouskos, F. Casati, N. Oertel, L. Mottola, F. J. Oppermann, G. P. Picco, K. Römer, and T. Voigt, "Process-based design and integration of wireless sensor network applications," in *Proceedings of the 10th international conference on Business Process Management*, ser. BPM'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 134–149. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32885-5_10
- [15] T. Rodrigues, P. Dantas, F. Delicato, P. Pires, L. Pirmez, T. Batista, C. Miceli, and A. Zomaya, "Model-Driven Development of Wireless Sensor Network Applications," in *9th International Conference on Embedded and Ubiquitous Computing, 2011*, oct. 2011, pp. 11–18.
- [16] N. X. Thang, M. Zapf, and K. Geihs, "Model driven development for data-centric sensor network applications," in *9th International Conference on Advances in Mobile Computing and Multimedia*, ser. MoMM '11. New York, NY, USA: ACM, 2011, pp. 194–197.
- [17] P. Boonma and J. Suzuki, "Model-driven performance engineering for wireless sensor networks with feature modeling and event calculus," in *Proceedings of the 3rd workshop on Biologically inspired algorithms for distributed systems*, ser. BADS '11. New York, NY, USA: ACM, 2011, pp. 17–24. [Online]. Available: <http://doi.acm.org/10.1145/1998570.1998574>
- [18] N. X. Thang and K. Geihs, "Model-driven development with optimization of non-functional constraints in sensor network," in *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications*, ser. SESENA '10. New York, NY, USA: ACM, 2010, pp. 61–65. [Online]. Available: <http://doi.acm.org/10.1145/1809111.1809128>
- [19] M. Mozumdar, L. Lavagno, L. Vanzago, and A. Sangiovanni-Vincentelli, "HILAC: A framework for hardware in the loop simulation and multi-platform automatic code generation of WSN applications," in *Industrial Embedded Systems (SIES), 2010*, july 2010, pp. 88–97.
- [20] K. Beckmann and M. Thoss, "A model-driven software development approach using OMG DDS for wireless sensor networks," in *Proceedings of the 8th IFIP WG 10.2 international conference on Software technologies for embedded and ubiquitous systems*, ser. SEUS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 95–106. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1927882.1927898>
- [21] G. Fuchs and R. German, "Uml2 activity diagram based programming of wireless sensor networks," in *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications*, ser. SESENA '10. New York, NY, USA: ACM, 2010, pp. 8–13. [Online]. Available: <http://doi.acm.org/10.1145/1809111.1809116>
- [22] N. Glombitza, D. Pfisterer, and S. Fischer, "Using state machines for a model driven development of web service-based sensor network applications," in *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications*, ser. SESENA '10. New York, NY, USA: ACM, 2010, pp. 2–7. [Online]. Available: <http://doi.acm.org/10.1145/1809111.1809115>
- [23] B. Akbal-Delibas, P. Boonma, and J. Suzuki, "Extensible and precise modeling for wireless sensor networks," in *UNISCON*, 2009, pp. 551–562.
- [24] M. Al Saad, E. Fehr, N. Kamenzky, and J. Schiller, "Scatterclipse: A model-driven tool-chain for developing, testing, and prototyping wireless sensor networks," in *Parallel and Distributed Processing with Applications (ISPA), 2008, Int. Symposium on*, dec. 2008, pp. 871–885.
- [25] M. Mozumdar, F. Gregoretti, L. Lavagno, L. Vanzago, and S. Olivieri, "A framework for modeling, simulation and automatic code generation of sensor network application," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, june 2008, pp. 515–522.
- [26] P. Jacso, "Google Scholar revisited," *Online Information Review*, vol. 32, no. 1, pp. 102–114, 2008.
- [27] D. Di Ruscio, I. Malavolta, H. Muccini, P. Pelliccione, and A. Pierantonio, "Developing next generation ADLs through MDE techniques," in *ICSE 2010*, 2010.
- [28] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang, "What industry needs from architectural languages: A survey," *IEEE Transactions on Software Engineering*, vol. 39, no. 6, pp. 869–891, 2013.
- [29] *ISO/IEC/IEEE 42010, Systems and software engineering — Architecture description*, ISO, December 2011.
- [30] L. Iovino, A. Pierantonio, and I. Malavolta, "On the impact significance of metamodel evolution in mde," *Journal of Object Technology*, vol. 11, no. 3, pp. 3:1–33, Oct. 2012. [Online]. Available: http://www.jot.fm/contents/issue_2012_10/article3.html
- [31] S. Simanta, K. Ha, G. A. Lewis, E. J. Morris, and M. Satyanarayanan, "A reference architecture for mobile code offload in hostile environments," in *MobiCASE*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, D. Uhler, K. Mehta, and J. L. Wong, Eds., vol. 110. Springer, 2012, pp. 274–293.