Formal Generic Frameworks for WSNs: a Review

Maissa Elleuch[1,2], Osman Hasan[2], Sofiène Tahar[2], and Mohamed Abid[1]

[1] CES Laboratory, National School of Engineers of Sfax, Sfax University
Soukra Street, 3052 Sfax, Tunisia
maissa.elleuch@ceslab.org
mohamed.abid@enis.rnu.tn

[2] Dept. of Electrical & Computer Engineering, Concordia University
1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada
{melleuch,o_hasan,tahar}@ece.concordia.ca

1 Introduction

In this report, I present a review of the generic formal frameworks for WSNs found in the literature. For each of them, a brief description followed by an analysis is given. At the end, I provide a general synthesis.

2 GLONEMO and LUSSENSOR

The GLONEMO and LUSSENSOR have been developed as parts of the PhD of L. Samper [28] who had worked in the Sychrone Team of the Verimag Laboratory of Grenoble in France [31].

2.1 Description of GLONEMO

The GLONEMO framework is mainly a formal simulation framework for WSNs where the WSN components are described in the same formalism [29] and implemented in ReactiveML [24]. The modeling formalism consists on interpreted automata that may be libeled with quantitative data. Such formalism is considered as expressive and quite general. The global model is made of separate models for the node, the medium, the observers which are other nodes, and the physical environment. One of the most important components which is the sensor node is viewed as the composition of a set of parallel processes describing the application, MAC and routing protocols. The MAC protocol chosen here is quite simple ; it is based on a preamble sampling while the routing is done by flooding. In addition, the environment is taken into account through the tool LUCKY [20] based on a constrained language.

The same formalism is executable allowing to simulate up to hundreds of nodes. The feasibility of the framework was shown through a very simple WSN whose goal to detect a radioactive cloud. The simulation process has quickly revealed that the flooding algorithm is not a good choice for the global model. Indeed, two nodes detect the same signal at the same time and then send the same alert message. This is mainly due to the modeling environment of the LUCKY tool that once modified into Poisson distribution, such situations will not occur. Finally, this point clearly demonstrates the importance of modeling the environment that has been intensively discussed in [30] .

2.2 Analysis of GLONEMO

Compared to other WSNs simulators, the GLONEMO framework is very different. On the one hand, GLONEMO take simultaneously into account the hardware of the node and its environment allowing an evaluation of energy. On the other hand, by using the same formalism, Samper promised to use the GLONEMO framework for formal verification purposes by its connection to some automated tools of formal verification such as [27] and [4]. But, nothing has been achieved in this direction. He has just analyzed the main steps needed for that which should be: reducing the model size by adjusting the abstractions of the components of the global model while showing their equivalence, then verifying global properties of safety on the model. Samper said that such connection should be considered only as a technical problem and the major difficulties that may be encountered should include the problem of state explosion. In addition, GLONEMO was limited in the modeling of the hardware which has been described by an automata libeled by values measured on [25]. This automata has been included into the MAC level. However, Samper pointed out that a more detailed energy model considering memory, processor or operating system consumptions, will be very complex.

2.3 Description of LUSSENSOR

As a part of the same thesis, a second simulator called LUSSENSOR inspired from GLONEMO, with a more detailed energy model, has been developed. The new energy model takes into account the behavior of the MAC protocol, the consumption of the radio, the CPU and memory. LUSSENSOR has been developed in LUSTRE [9] which is a deterministic language suited for WSNs modeling. The MAC protocol of the global model is a preamble sampling with a random back off procedure to avoid collisions. The routing protocol chosen here is direct diffusion in 2 phases [19]. The application code is a very simple algorithm that sends the detected values to a base station.

2.4 Analysis of LUSSENSOR

The LUSSENSOR framework has the advantage to give a very good model of the hardware so that the energy can be evaluated more accurately. Nevertheless, its effectiveness has not been shown on a concrete example. Indeed, the simulation was very slow even for a WSN of 10 nodes especially when compared to GLONEMO. Hence, the main advantage of LUSSENSOR is to provide good opportunities for formal verification: the LUSSENSOR syntax is close to the formal verification tools available at Verimag such as [27] and [4]. There are two other major drawbacks of the LUSSENSOR framework. Similar to GLONEMO, the energy measures were very specific [25]. Added to that, the modeling of randomness was impossible because the LUSTRE language is deterministic. Thereby, if a component needs random values, the global model has to be connected to external generator which will give explicit random values. Such a modeling is not so accurate and can not in any case give realistic results.

In conclusion, the GLONEMO and LUSSENSOR approaches seem to be very innovative especially by the idea to integrate formal tools for analysis. Samper has indicated that the global model of the GLONEMO framework has been enriched later so that it has been successfully used in the industry. Nevertheless, no real connection with a formal tool has been done even after the end of his thesis. These two frameworks are not open source.

3 Slede: an automatic framework for the verification of security in WSNs

The Slede framework has been developed by Y. Hanna [16] who is still a PhD student in the Laboratory of Software Design at Iowa State University in the USA.

3.1 Description

The traditional approach for the verification of security protocols is based on intrusion patterns written manually. Such approach is both long and tedious. To address these limitations, SLEDE is a framework for the automatic verification of implementations of security protocols for WSNs [10]. Specifically, this approach includes:

- A technique for extracting the model from its implementation in nesC [8],
- A technique for the generation and decomposition of the extracted model with models of intrusion,
- A technique for the verification of security properties by generating counter-examples demonstrating the violation of properties.

Fig. 1 illustrates the general approach of SLEDE. Thus, from the nesC implementation, SLEDE is able to automatically extract the model of the protocol. At this stage, SLEDE needs information about the topology and the structure of exchanged messages. At the same time, SLEDE generates specific patterns of intrusion from the protocol specification. For that, it uses an intruder template library. Finally, the whole framework tries to verify, using the model checker SPIN [12], the security properties of the model. If any of these properties is violated, SLEDE generates counter-examples which will be translated to nesC.

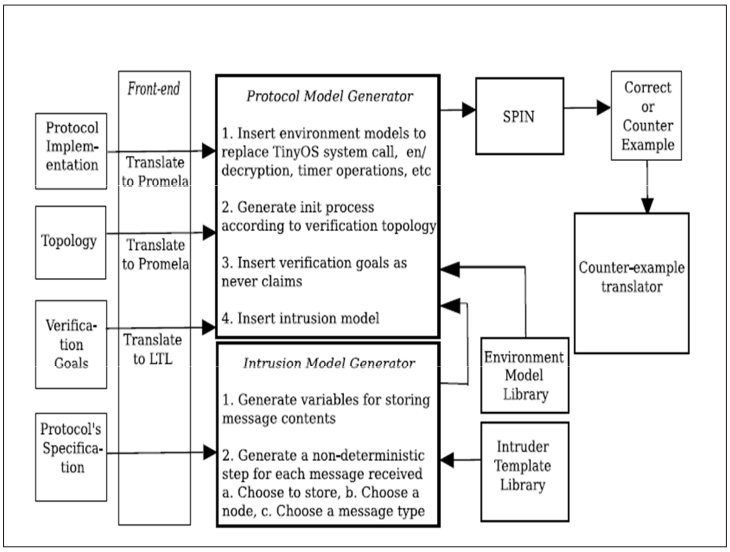
Protocol Implementation
Topology
Verification Goals
Protocol's Specification
Front-end
Translate to Promela
Translate to Promela
Translate to LTL
Protocol Model Generator
1. Insert environment models to replace TinyOS system call, en/decryption, timer operations, etc
2. Generate init process according to verification topology
3. Insert verification goals as never claims
4. Insert intrusion model
Intrusion Model Generator
1. Generate variables for storing message contents
2. Generate a non-deterministic step for each message received a. Choose to store, b. Choose a node, c. Choose a message type
SPIN
Correct or Counter Example
Counter-example translator
Environment Model Library
Intruder Template Library

Fig. 1. Overview of the Slede framework [10].

3.2 Analysis

SLEDE has the advantage of reducing the verification efforts of security implementations at the expense of learning a specification language that seems to be very close to implementation language. The effectiveness of SLEDE has been shown on two common security protocols where known security flaws has been detected. Nevertheless, SLEDE is also limited by the state space explosion problem. To overcome this major limitation, the developer proposed the use of parameterized model checking. At the end, despite the soundness of the approach, it is incomplete since all security flaws cannot be detected for a given implementation.

4 CaVi

The CaVi tool has been developed by A. Fehnker [15] who is a researcher at the National ICT Australia in the Managing Complexity research theme. The developer of the Castalia simulator for WSNs, A. Boulis [14], contributed also in CaVi.

4.1 Description

CaVi [7] is a tool resulting from the combination of a WSNs simulator called Castalia [26] and the model checker PRISM. CaVi should have the advantage to provide a graphical interface in which the simulation results of a given model from Castalia can be then exhaustively analyzed in PRISM given a fixed property (Fig. 2). In particular, for now, a realistic channel behavior has been introduced into formal models of PRISM. Then, it is possible to capture the effect of topology by computing multi-hop reception probabilities within PRISM. The developers advocated that PRISM outperforms Monte Carlo simulation when computing such probabilities by giving exact and not average probabilities. Nevertheless, no further investigation for connecting PRISM to CaVi has been done until now. Task like automatic translation of PRSIM models from CaVi remains as future work.

4.2 Analysis

The CaVi tool has been limited to the performance evaluation of WSNs. In the literature, we don't find a lot of details about CaVi. The last publication dates back to 2008 [3] and is a short paper summarizing what has been yet done but no new achievement. Each of the tools Castalia and PRISM taken separately seems to be very mature. However, their combination into the Cavi tool does not seem to be so. Indeed, given the capabilities of probabilistic model checking, there are many other aspects to be included in CaVi. In addition, no real case study on WSNs has been made to show the effectiveness of this tool.

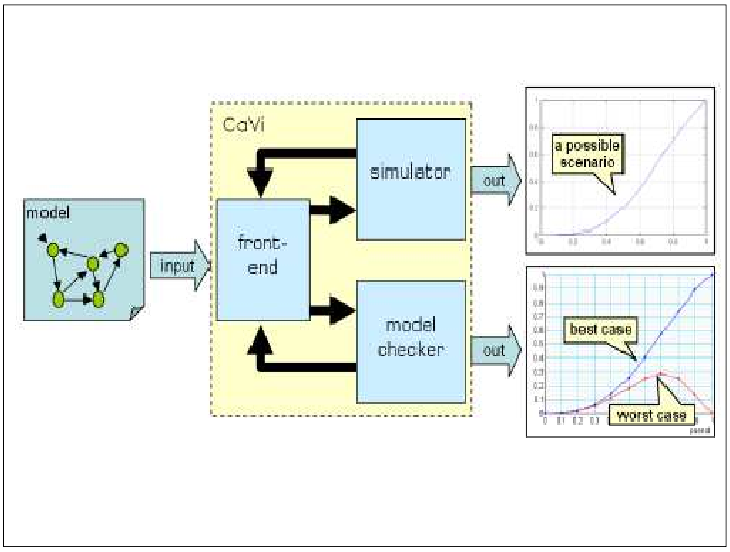

Fig. 2. Overview of CaVi tool [7].

5 TEPAWSN

In [23], the authors give the proposal of the TEPAWSN tool for the qualitative and quantitative performance evaluation of WSNs and the estimation of the effective energy.

5.1 Description

The TEPAWSN tool should use the PAWSN (Process Algebra for Wireless Sensor Network) language which combines the features of classical process algebra as the sequential and parallel composition, with time and non-deterministic behavior and/or probabilistic WSNs including energy issues. The general approach of the TEPAWSN tool is presented in the figure below. It consists in describing the WSN in the PAWSN formalism. After that, a conversion step follows in order to translate the described model into the language of the target tool (PRISM, PPTA, nesC ...) for simulation, formal verification or energy analysis. This conversion will be ensured by appropriate tools such as PAWSN2PPTA, and PAWSN2nesC PAWSN2PRISM.

5.2 Analysis

The TEPAWN tool should have the advantage to describe the target WSN in a single and expressive formalism. Moreover, it benefits from existing analysis tools for simulation, model checking or energy evaluation. The main work should focus on the development of good conversion tools. However, TEPAWN is still at the specification stage and no achievement could be found so far. Finally, the language PAWSN used was briefly described in [23] and no proof for its effectiveness was given.

6 The PVS framework for WSNs

The PVS framework for WSNs has been developed by C. Bernardeschi [13], P. Masci [18], and H. Pfeifer [17].

6.1 Description

In [1], a simulation and analysis framework for WSNs algorithms within the PVS system is proposed. PVS provides at the same time simulation through a ground evaluator [6] and formal verification within a theorem prover. The approach can be applied at the early stage of the development process to consolidate the algorithm design. A WSN algorithm is modeled by assembling a collection of components. In this framework, the main components considered are the nodes, the network structure, communication primitives and protocols. The communication primitives are functionalities for communication between nodes like the forwarding, injection and dropping of packets, while the protocols are specified

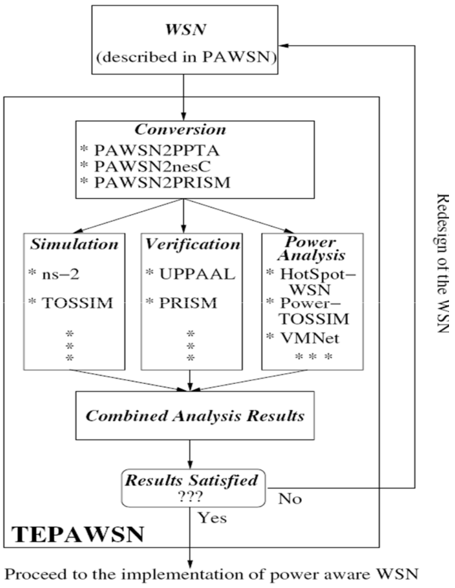

Fig. 3. Overview of TEPAWN tool [23].

so that they can use services installed on nodes. Services include the packet logger, the receive buffer, the node scheduler and the clock. Each component is described by a PVS model taken from the corresponding theory. Different versions of the same component are available so that it is possible to specify and analyze WSN algorithms under several perspectives and at desired level of abstraction. A very abstract theory consists on the declaration of attributes type and interface functions. More detailed theories can be derived from the abstract definition by specifying the behavior of interface functions, and by extending types.

In the following, the main WSN components modeled by the PVS framework for WSNs are detailed. The connectivity of the network is described with direct graph without self-edges. To reuse existing theories, definitions are built on the top of the library of directed graphs developed by NASA [5]. A sensor node is identified by a natural number. An intermediate function is specified to identify, for each node, the set of neighboring nodes. Both ideal and loss links are modeled, and topology changes can be used to model node mobility.

Depending on the algorithm specification and on the property of interest, services can be installed on a single node, on a group of nodes, or on the entire network. Many services have been implemented such as:

- packet logger, which stores statistics about sent and received packets,
- receive buffer, which models the buffer where packets sent by other nodes are stored,
- energy consumption, which evaluates the energy spent by nodes,
- routing, which provides the basic definitions for building routing tables, spanning trees and paths between nodes,
- node scheduler, which gives the sequence of nodes that execute the algorithm (e.g., round robin, or random).

A structure called networkstate shows the state of a given network through the state of its different nodes which is mainly described by the allocated service. The communication primitives are also taken into account. In particular, the packet format is specified in PVS in a separate theory. Communications are modeled through three main functions. The algorithm to analyze is specified as a cyclic procedure executed on a generic node. In [1], the authors showed how to specify the flooding algorithm [11] within the PVS framework for WSNs. Once specified, the algorithm can be analyzed at different level of abstractions by showing or hiding the properties of interest. For example, the energy consumption can be analyzed by importing the corresponding theory of the network state.

The feasibility of the developed framework has been shown on the Surge algorithm which is a widely used routing protocol for WSNs included in TinyOS [22]. The authors evaluate the receive queue size, the energy consumption and the robustness to topology changes on a network of maximum 25 nodes with different topologies. By inspecting the execution traces, they were able to detect a potential problem of infinite loops of routed packets in the algorithm specification.

After that, the framework has been enriched by theories expressing dynamic scenarios in WSNs like nodes mobility, link quality changes, and the automatation of the generation of routing tables. The mobility pattern used here is the random walk which moves the mobile node n times. It is specified through a recursive function. For lossy channels, the quality can change with a uniform probability Pc. Then, routing table can be generated by instantiating this probability Pc. Hence, In order to show the practical effectiveness of the enriched framework under dynamic scenarios, a second case study has been done on the reverse path forwarding (RPF) algorithm [2]. When simulating the RPF, the probability has been instantiated with four different values. Added to that, the main package has been attached to a generator of pseudo-random numbers to generate random sets of nodes. Considering networks of 64 nodes placed on a grid with 8 columns, the authors have evaluated the delivery ratio and the overhead due to duplicates of the RPF algorithm. They have also simulated a scenario where a mobile station caused the modification of the routing table. Finally, a property of correctness has been formally verified. It says that "if the routing table is correct and static, then exactly one copy of the broadcast packet sent by the base station will be delivered to all nodes in the network". The property has been formalized by a theorem which has been successfully verified within the PVS theorem prover. This proof has needed many intermediate lemmas.

Question: From [2], what do they mean by "the formal proof was done by an induction on the execution traces of RPF, i.e., on sequences that start with the initial state and apply the RPF transition function to generate subsequent states." Do they verify their theorem on the execution trace of simulation and then by theorem proving?

6.2 Analysis

The PVS framework for WSNs is very interesting. Indeed, this framework can be used simultaneously for formal specification, automated simulation and verification of the behavior of a protocol designed for WSNs. The practical effectiveness has been shown on two case studies. The major drawback of the PVS framework is the probability modeling which was not so accurate. First, the randomness of numbers was taken into account through an external theory which generates pseudo-random numbers. No further detail has been given on this theory. Second, when considering that the link can change with a uniform probability Pc, the authors had just set this probability. Finally, a weakness in the second case study can be noticed. Indeed, the property of interest has been formally verified for static routing table while the authors have clearly pointed out that routing table within real-world deployments are most of the time dynamic.

7 Synthesis

Although, the idea of developing a generic formal framework for WSNs is quite promising, it is not really widespread. What is rather more common is the development of simulators dedicated to WSNs. While the existing simulators have been always limited in terms of modeling the hardware and the environment, the explored frameworks try to give a more complete solution. The frameworks found in this context such as GLONEMO and CaVi can be considered as formal simulation frameworks but not as generic. First, they are formal because they use a single and expressive formalism in the description of the WSN. This feature can be considered as the main start point for a potential formal analysis. Second, these frameworks have given the general approach and promised to make connection with some formal analysis tools but none of them has made such connection. They were limited to simulation. For GLONEMO, no connection with formal tools is made even after the end of the PhD of Samper. In the case of CaVi, for each simulation batch, it is just possible to compute the exact multi-hop reception probability using PRISM. No real connection with PRISM has been done by, for example, verifying properties of the simulated model. For the TEPAWSN tool, it is still at the specifications stage. On the other hand, they cannot be considered as generic since the models used for hardware, MAC and routing protocols are very simplified. Other generic frameworks such as SLEDE taking into account only one aspect and inspecting directly the implementation seem to be more affordable, realistic and effective. Moreover, the effectiveness of the tool was shown on two case studies.

The most interesting work found in this context is for sure the PVS framework for WSNs which can be considered simultaneously as formal and generic. It is formal because the model is written in a formal semantic allowing fast performance evaluation within the PVS simulator. Moreover, the framework is generic enough since each component, described in a separate theory, can benefit from different model versions so that the abstraction level can be varied. Once specified formally in PVS, it is possible to formally verify properties of the built model; a novelty compared to previous frameworks which promised formal verification but remained limited to simulation. The applicability of the framework has been shown on two case studies which are the Surge and the RPF algorithms designed for WSNs. The formal verification has been successfully done only on the second algorithm. Finally, it is important to note that the PVS framework is open source. The major drawback of the PVS framework is the inaccuracy of the randomness modeling: the numbers generated were pseudo-random and the uniform probability was just fixed. No further indication has been given about the external random generator or the uniformity of the probability. Nevertheless, given the probabilistic features of WSNs, it becomes essential to include an accurate modeling of probability so that the reality can be reflected. Within, the probabilistic framework developed in the HOL theorem prover, we can overcome all the drawbacks cited above to develop a generic formal framework for WSNs and verify interesting properties on the built model.

Questions: Related to formal verification

1. Is the idea of a generic and formal framework for WSNs is not in itself very ambitious because most of the tool attempts in this direction were not really

achieved?

2. Back to the GeNoC work (Master thesis), the translation of the generic definitions in ACL2 was possible thanks to the encapsulation principle [21]. In general, this principle lets the user introduce constrained function without explicit definition. When the encapsulated function f is admitted to ACL2, the theory is extended by the following event ”the function f is constrained by the axiom ϕ”. Thus, the function f does not have an explicit definition but we know that f has to satisfy the property ϕ. Consequently, a function g is an instance of the encapsulated function f, if and only if, g satisfies all the constraints defined for f at the generic level.
I'm asking if there is a similar aspect in HOL?

Questions: Related to WSNs

1. If the idea of building a generic and formal framework for WSNs is adopted, what should be the most important WSNs components to model so that at least a minimal but complete WSN model can be described through the framework?

References

1. C. Bernardeschi, P. Masci, and H. Pfeifer. Early prototyping of wireless sensor network algorithms in pvs. In *Proceedings of the 27th international conference on Computer Safety, Reliability, and Security*, SAFECOMP '08, pages 346–359, Berlin, Heidelberg, 2008. Springer-Verlag.
2. C. Bernardeschi, P. Masci, and H. Pfeifer. Analysis of wireless sensor network protocols in dynamic scenarios. In *Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, SSS '09, pages 105–119, Berlin, Heidelberg, 2009. Springer-Verlag.
3. A. Boulis, A. Fehnker, M. Fruth, and A. McIver. Cavi – simulation and model checking for wireless sensor networks. In *Proceedings of the 2008 Fifth International Conference on Quantitative Evaluation of Systems*, pages 37–38, Washington, DC, USA, 2008. IEEE Computer Society.
4. M. Bozga, S. Graf, and L. Mounier. If-2.0: A validation environment for component-based real-time systems. In *Proceedings of CAV'02*, volume 2404, pages 343–348. Springer-Verlag, 2002.
5. R. W. Butler and J. A. Sjogren. A pvs graph theory library. Technical report, NASA Langley Technical Report Server, Hampton, Virginia, 1998.
6. J. Crow, S. Owre, J. Rushby, N. Shankar, and D. Stringer-Calvert. Evaluating, testing, and animating pvs specifications. Technical report tr-2004-6, Computer Science Laboratory, SRI International, Menlo Park, CA, 2001.
7. A. Fehnker, M. Fruth, and A. McIver. Graphical modelling for simulation and formal analysis of wireless network protocols. In *Proc. Workshop on Methods, Models and Tools for Fault-Tolerance (MeMoT'07) at the 7th International Conference on Integrated Formal Methods (IFM'07)*, pages 80–87, Berlin, Heidelberg, 2007. Springer-Verlag.

8. D. Gay, P. Levis, R. von Behren, M. Welch, E. Brewer, and D. Culler. nesc the language: A holistic approach to networked embedded systems. In *Proceedings of the 2003 conference on Programming Language Design and Implementation*, pages 1–11, New York, NY, USA, 2003. ACM.
9. N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud. The synchronous dataflow programming language lustre. *Proceedings of the IEEE*, 79(9):1305–1320, September 1991.
10. Y. Hanna, H. Rajan, and W. Zhang. Slede: a domain-specific verification framework for sensor network security protocol implementations. In *Proceedings of the first ACM conference on Wireless network security*, WiSec '08, pages 109–118, New York, NY, USA, 2008. ACM.
11. W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings Intl. Conf. on Mobile Computing and Networking*, pages 174–185. ACM Press, 1999.
12. GJ. Holzmann. The model checker spin. *IEEE Trans. Softw. Eng.*, 23(5):279–295, 1997.
13. Bernardeschi homepage. http://www.iet.unipi.it/c.bernardeschi/.
14. Boulis homepage. http://www.nicta.com.au/people/boulisa.
15. Fehnker homepage. http://www.cse.unsw.edu.au/˜ansgar/.
16. Hanna homepage. http://www.cs.iastate.edu/˜ywhanna/.
17. Holger homepage. http://hp.sibilatores.de/en/hp/.
18. Masci homepage. http://www.eecs.qmul.ac.uk/˜masci/.
19. C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of MOBICOM*, pages 56–67. IEEE Computer Society, 2000.
20. E. Jahier and P. Raymond. The lucky language reference manual. Technical report tr-2004-6, Verimag, 2005.
21. M. Kaufmann and J. S.Moore. Structured theory development for a mechanized logic. *J. Autom. Reason.*, 26(2):161–203, 2001.
22. P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *Proceedings of the Intl. Conf. on Embedded Networked Sensor Systems*, pages 126–137. ACM Press, 2003.
23. K.L. Man, T. Krilaviius, T. Vallee, and HL Leung. Tepawsn-a formal analysis tool for wireless sensor networks. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 1:24–26, October 2010.
24. L. Mandel and M. Pouzet. Reactiveml: a reactive extension to ml. In *Proceedings of the 7th ACM SIGPLAN international conference on Principles and practice of declarative programming*, PPDP '05, pages 82–93, New York, NY, USA, 2005. ACM.
25. Motorola Freescale 2005 Motorola MC13192 data sheet. freescale.com/files/rf if/doc/data sheet/mc13192ds.pdf.
26. H.N. Pham, D.Pediaditakis, and A. Boulis. From simulation to real deployments in wsn and back. In *WOWMOM'07*, pages 1–6, 2007.
27. C. Ratel, N. Halbwachs, and P. Raymond. Programming and verifying critical systems by means of the synchronous data-flow language lustre. In *Proceedings of the conference on Software for citical systems*, SIGSOFT '91, pages 112–119, New York, USA, 1991. ACM.
28. L. Samper. *Modeling and Analysis of Sensor Networks*. PhD thesis, INPG, France, 2008.

29. L. Samper, F. Maraninchi, L. Mounier, and L. Mandel. Glonemo: Global and accurate formal models for the analysis of ad hoc sensor networks. In *Proceedings of the First ACM International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense'06)*, New York, USA, 2006. ACM.
30. L. Samper, F. Maraninchi, L. Mounier, L. Mandel, E. Jahier, and P. Raymond. On the importance of modeling the environment when analyzing sensor networks. In *Proceedings of International Workshop on Wireless Ad-Hoc Networks 2006 (IWWAN 2006)*, pages 835–841. IEEE Press, 2006.
31. Synchrone team. http://www-verimag.imag.fr/synchrone,30.html?lang=en.