# Proof-based formal methods for WSN development with Simulation Approach

Adisak Intana, Michael R. Poppleton, and Geoff V. Merrett

Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK
{ai1n10,mrp,gvm}@ecs.soton.ac.uk

Software engineering (SE) plays a fundamental role in wireless sensor network (WSN) development due to the appearance of WSN adoption over wide range of the real-world physical applications. The WSN software development process focuses on "low phases" of development such as programming, testing and deployment. In current practice, programmers decompose the problem and then develop code for sensor nodes based on the functionality provided by the operating system platform of choice before integrating with low level hardware [1]. The current WSN development process is performed under very limited OS or development resource available [2]. *Simulation testing*, *emulation testing* and *laboratory testbed techniques* are the main approaches to analyse and evaluate the correctness and performance of network algorithms and protocols [3]. However, since WSNs have been applied to safety-critical application domains from healthcare to military applications, the high-level abstraction and verification and validation techniques have become an important key for current WSN development [2]. It is quite evident that they lack SE methods, techniques and tools which support "high phases" of current practices on WSN development such as requirements and architecture specification [4,5,6].

The vision of this research is the strengthened proof- and animation-based verification at high phases of WSN development process by integrating formal methods with existing simulation technologies. Formal modelling and proofs can guarantee the basic functionality of a WSN system, whereas animation technology can validate system behaviours. This leads to an increase in the programmer's confidence about the correctness of their protocols and algorithms before core simulation models are generated from formal models.

This research proposes a hybrid verification and validation approach for the "high phases". The proof-based formal methods like Event-B are combined together with simulation to increase the quality of WSN development. An environmental application from *SensorScope* project [7] was chosen to demonstrate our case study work for such integration approach. *MintRoute*, together with *S-MAC* protocol is simulated with the connectivity failure scenarios using the *MiXiM* simulation tool. We found that simulation shows the presence of faults such as the routing loop problem but cannot guarantee that our algorithm can be performed without loop. Thus, RODIN [8] was used for Event-B modelling and proof obligation (PO) purposes in order to tackle the absence of such faults from simulation. By using Event-B, MintRoute protocol was modeled through refinement steps.

We started developing our Event-B model with a very simple abstract model, **Initial Model (M0)**, for transmitting a data packet from source nodes to a sink via the network abstractly. Then, in **First Refinement (M1)**, the neighbour nodes were considered for forwarding a packet hop by hop from source nodes to a sink. In **Second Refinement (M2)**, neighbourhood discovery and link quality estimation mechanisms for MintRoute protocol were taken into account. **Third Refinement (M3)** introduced parent selection and route broadcast mechanisms for MintRoute protocol. The safety properties for the route tree such as "no-loop" were introduced and proved in this refinement. *ProB* was applied with node failure scenarios to validate model behaviours by tracing step by step.

Our result shows that Event-B can manage complexity of the application through refinement. Furthermore, the consistency and correctness of encoded functional requirements such as network algorithms and safety properties are verified by POs. The absence of looping problem also has been proved by discharging POs. Further formal V&V regarding deadlock, safety and liveness properties will be considered in our current Event-B network models. We also discovered the correspondence between Event-B and C++ (the simulation language). For example, the constant and variable declaration concepts of Event-B ($minCost \in \mathbb{N}$) is similar to that of C++ (*int minCost*;). The algorithm control mechanism of C++ such as sequence, branch and loop can be indicated by groups of sub-events from atomicity refinement. Our aspiration is automatic tool-supported partial translation of formal to simulation models. This remains work for the future.

## References


1. L. Mottola and G. P. Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 19:1–19:51, 2011.
2. G. P. Picco, "Software engineering and wireless sensor networks: happy marriage or consensual divorce?," in *FoSER '10*, pp. 283–286, ACM, 2010.
3. M. Korkalainen, M. Sallinen, N. Kärkkäinen, and P. Tukeva, "Survey of wireless sensor networks simulation tools for demanding applications," in *ICNS '09*, pp. 102–106, IEEE Computer Society, 2009.
4. J. A. Stankovic, "Research challenges for wireless sensor networks," *SIGBED Rev.*, vol. 1, pp. 9–12, July 2004.
5. 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 '12*, pp. 1–7, 2012.
6. R. Shimizu, K. Tei, Y. Fukazawa, and S. Honiden, "Case studies on the development of wireless sensor network applications using multiple abstraction levels," in *SESENA '12*, pp. 22–28, 2012.
7. G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Out-of-the-box environmental monitoring," in *IPSN '08*, pp. 332–343, IEEE Computer Society, 2008.
8. J.-R. Abrial, M. J. Butler, S. Hallerstede, and L. Voisin, "An open extensible tool environment for event-b," in *ICFEM*, pp. 588–605, 2006.