Ensuring Feasibility of Wireless Sensor Networks

Simone Campanoni, William Fornaciari
Dipartimento di Elettronica e Informazione (DEI)
Politecnico di Milano
P.zza L. Da Vinci, 32 - 20133 Milano, Italy
{campanoni, fornacia}@elet.polimi.it

***Abstract*—A comprehensive WSNs (Wireless Sensor Networks) design methodology should deserve enough effort to select a suitable set of sensors with a proper spatial distribution, in order to ensure a correct monitoring of the parameters relevant for the application. The goal of this paper is to show how it is possible *a priori* ensuring that a given WSN is actually capable to satisfy the application requirements. In addition to this, the flexibility of the approach is also demonstrated by evaluating the sensitivity of the WSN performance to parameters like sensors distribution, observation time and the events to be analyzed. This paper presents the overall design methodology implemented in SWORDFISH (Sensor netWORks Development Framework Integrating Simulation and Hardware optimization) and some representative case studies.**

I. Introduction

During last decade, the use of the Sensor Networks is gaining importance, especially as a consequence of the seed constituted by the wireless architectures designed at Berkeley, already landed on the market [1]. Such solution probably has been the first pioneer (successful) attempt to target a wide market, including low end purposes, by providing both the bare hardware and some middleware to simplify the synthesis of applications. Building a WSN is going to become like composing COTS (Component Off The Shelf).

Despite such simplifications, many other questions are still open or partially neglected, like optimization of overall costs (sensors, communication infrastructure, deployment, …), feasibility analysis to understand suitability and effectiveness of the WSN against real application goals, lifetime (especially in the case of battery operating sensor nodes), robustness, etc [1] [2].

The main lack is the missing of an overall analysis and design framework, to enable a quantitative evaluation of the above properties, taking into account not only the networking-related issues or the distributed software system itself, but also the cross relations existing among the network topology, the nodes, the environment where the WSN is embedded and the events to be monitored, namely the real and comprehensive *functional* goal of the WSN.

Since a few years, in literature appeared a number of proposals regarding simulation and deployment of WSNs; some of the more mature and publicly available results are listed in [1-12]. Each of these proposals addresses with meaningful results some specific simulation or implementation level aspects of WSN analysis, covering hardware, software and networking. Unfortunately, from the best of our knowledge, up to now none is addressing with a proper and formal extent the capability of the network to capture the events to be monitored, since the primary focus is frequently related to the optimization of the cost or to verify other properties like power consumption, robustness of the connection layer or the analysis of the middleware-level models of computation.

The scope of the work here presented is a wide class of applications where, in addition to the typical monitoring capabilities, it is also required a prompt highlight of the occurrence of particular events. Under these assumptions, our methodology to tackle the problem of designing a sensor network requires to:

- **specify** the characteristics of the events of interest;
- **select** a proper set of sensors tailored to catch such events;
- **embed** the sensors in the environment in a way to formally ensure the capturing of the desired events while optimize some design goals.

The objective is first of all to make sure *a priori* that it exists a feasible solution to the sensing problem with the accuracy required by the application. Then, by exploiting the capabilities of the SWORDFISH optimization engine, it is possible to derive the WSN by refining the architecture according to design constraints and user's goals.

The paper is organized as follows. Section II summarizes the overall architecture of SWORDFISH. Section III discusses the models of the events to be recognized and the design flow to create a WSN ensuring that all the events can be sensed. Some of the capabilities of SWORDFISH are discussed in Section IV, where it is shown how it is possible to explore the design space taking into account both abstract and functional requirements. Concluding remarks are drawn in Section V.

II. The Design Flow

The general architecture of SWORDFISH is depicted in Fig. 1. It is composed of a set of modules allowing the users to describe the actors (sensors, network, events, and environment) and the design goals of the systems (properties of the network and optimization parameters). The overall framework is encapsulated in a graphical user interface connecting all the different modules, whose role and main characteristics are outlined in the following.

978-1-4244-1708-7/08/$25.00 ©2008 IEEE

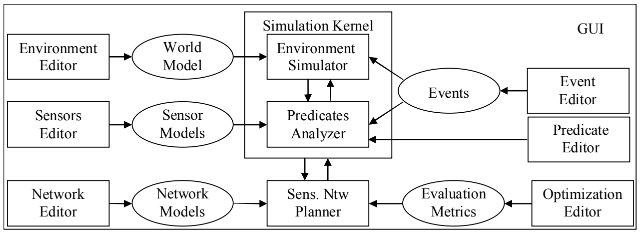

Figure 1. Top-level architecture of SWORDFISH.

Environment Editor. This module allows defining a model of the environment where the WSN will be embodied, with graphical views of the associated physical parameters (e.g., temperature, humidity, 3D-spatial representation, obstacles, …) and the possibility to specify constraints such as position and type of some sensors, if relevant for the users.

Sensor Editor. It is the mean to obtain the analytic representation of the sensing nodes, which is a modeling of the relation existing between the sensed physical parameters and the signal produced. The model of the node includes additional information like cost, type of sensors, energy consumption, accuracy, speed, etc.

Network Editor. In addition to the node features, a model of the available connection channels among nodes is specified. This model can cover both wired and wireless links, although in our first implementation we focused on wireless only.

Predicate Editor. This editor allows the user to specify via logic formulas the properties to be verified in the case a given event occurs. This is of paramount importance to verify that a WSN is actually capable to argue if an event is recognized, or, dually, to select the proper set of sensors to recognize the events. Such a concept more abstract and powerful then a simple measurement-based analysis.

Event Editor. The purpose of this editor is to support the description of the events to be captured in terms of variation of some physical parameters to be sensed, along with their timing characteristics. These models are flexibly implemented via software plugins.

Simulation Kernel. It is the engine which, based on a simulation of the event occurring, modifies the configuration of the world model accordingly. This allows feeding the sensor node models with the real (location aware) data of the world, including their dynamics. Hence, both the physical parameters of the environment and the events to be monitored can be jointly modeled and verified by the Predicate Analyzer (Fig.1)

Optimization Editor. It is an editor allowing the designer to specify and tune the goal functions and the formal model of the network properties/constraints.

Planner. This is the main module for both verification and network design. It allows to formally verifying that a given WSN is able to capture a set of events as well as to support the building and optimization of the overall network according to the selected policies and goals.

The architecture of SWORDFISH is conceived to support the users during the system-level design of the application. In particular, the following problems can be addressed.

Verification. The goal is to determine the occurrence of a set of events (e.g., fire in a defined region, temperature and humidity over a certain threshold for a time window, etc.) by exploiting the *potential* of a given sensor network.

Sensitivity Analysis. Evaluation of the impact of some variation of sensors, environment and network properties, onto the performance of a WSN. Examples are fault tolerance w.r.t. sensors and network errors, effect of sensor aging or moving of their location, influence of the observation time, etc.

Design/Planning. Given a set of events and some constraints/goals, the task is to discover the optimal sensor network capable to identify the events while maximizing a user-controlled goal function.

The focus of this paper is on the sensitivity analysis and on some planning strategies enabled by SWORDFISH. Based on the application requirements, the first steps for the user are formally defining the events to be captured and possibly some optimization goals/constraints. Network properties and sensor behavior can be also specified, in the case of default settings are not considered suitable. According to the existing model of the environment, the events are then "fired" to get a profiling of the evolution of the physical parameters corresponding to the events. Such results are then used as a testbench to compare the performance of alternative WSNs in terms of sensing capabilities. The predicate analyzer and the selected optimization goals are extensively used by the network planner to explore the design space. Useful information for optimization can be gathered by analyzing the sensitivity of the network over the variation of parameters like observation time or clustering of sensors, as shown in Section IV.

III. WSN ARCHITECTURE DESIGN

The model of the environment is 3-D, so that each point is represented using (x,y,z) coordinates belonging to a user-defined grid. Before starting the exploration of the WSN design space, there are three preliminary steps to be carried out: i) definition of the *purpose* of the network; ii) identification of the *benchmark*; iii) modeling of the *hardness* to recognize physical parameters corresponding to an event.

The first activity is the definition of an overall **Sensing Goal** (SG) for the WSN, that is a multi-value logic formula composed of some predicates Pr (implemented via plugins), each corresponding to an event. For example Water(x,y,z,magn,trend) is a plug-in modeling the presence of *water* in the point (*x,y,z*), starting from a given *magnitude* and with a specified *trend* over the time. A predicate Pr is an instance of Water applied to a specific point. A catalog of plugins (e.g., Fire, Water, Humidity…) is available, but its extension is straightforward. An example of sensing goal is (1).

SG=Water (0,1,2, 20, const) AND Water (3,3,5, 10, const) (1)

Such SG means that the WSN has the goal to discover the concurrent presence of the events of having a certain amount (20 and 10) of water in two points (0,1,2), (3,3,5) of the environment.

The second step is the characterization of the changing in the environment whenever the events occur, namely the identification of a testbench to evaluate the WSN performance. To this purpose, based on the (user defined) *fp* sampling rate of the environment simulator, a profiling stage is triggered by firing each of the defined events, namely running the Pr-related plugins. At the end, ∀(x,y,z), and ∀ Pr of SG, all the data patterns are obtained.

There are at least other two problems the designer has to face with during WSN design. The first concerns the selection of the type of sensor, while the second is the sensors placement. In fact, the target is to discover a positioning of the sensors, maximizing the capability of the WSN to recognize the events, i.e. maximizing the SG. The former question impacts mainly on the feasibility of designing a WSN capable to recognize the events encompassed by the SG. The latter is related to the dissemination of sensors in order to enhance their possibility to satisfy the Pr composing the SG, i.e. improving the performance of the system.

Another important aspect that is not discussed here due to lack of space is related to the clustering of the single sensing capability onto a set of nodes to minimize development and deployment costs, with an acceptable performance degradation.

In the current implementation of SWORDFISH, we followed an approach allowing obtaining results in the order of seconds, so to actually enable sensitivity analysis, whose fully discussion is beyond the scope of this paper. In the following, the main benefit of sensitivity analysis are addressed only though some representative examples.

Our first concern in the design flow is ensuring that a solution to the SG can exists, using a proper set of sensors that is incrementally built up and significantly optimized by sharing sensors among the set of Pr (specified in the SG) to be verified. Then, this set of candidate sensors are placed in the environment taking into account the information coming from a configurable *hardness* function. In such a way it is guaranteed to obtain a WSN formally satisfying the SG with a quasi-optimal cost, with runtimes in the order of a few seconds.

As far the positioning of the sensors is concerned, we defined a *hardness* function Hard(x,y,z,Pr) modeling the difficulty in evaluating Pr in a given point (x,y,z).

Hard(x,y,z,Pr)= Hs(PPr,t)/ C{(PPr,t), Pr} (2)

Where (calling PPr the "profiling output" of Pr, i.e. the data pattern associated to Pr obtained during the initial profiling):

- Hs(PPr,t): depends on the type of sensor (corresponding model) and relates to the difficulty to recognize the event Pr within the time frame of a profiler sampling rate (1/fp). For example for a slow temperature sensor can be hard (or even impossible) recognizing T-ramps moving faster than its cutting frequency.
- C{(PPr(x,y,z), Pr} is the confidence to infer the truth of Pr based on the sequence of the physical variations defined via PPr.

Of course, any positioning strategy for the sensors attempts to place the sensor where *Hard* is low, i.e. where it is easier and reliable recognizing the Pr composing SG.

To better explain how the selection of the proper sensors take place, let us consider a simple example: three sensors (S1, S2 and S3) have been identified valuable for the four predicates P1-P4, and P4 is not yet covered by any sensor (see Tab.I). Our goal is to ensure the selection of a proper set of sensors capable to cover all the predicates composing the SG.

The implemented strategy is quasi-optimal and in this case it search for a sensor among S1-S3 to sense (cover) also P4, such that its sharing produces the minimum impact onto the overall satisfying of SG, as already obtained through P1-P3.

To support such optimization process, the operators of the SG logic formula are mapped onto a derivable expression. In particular AND and OR logic operators have been mapped onto "+" and "*" algebraic operators. In such a way it is simplified the analysis of the influence of SG w.r.t. each of the Pr composing it, by simply considering its partial derivative.

More formally, it is selected the Si to be assigned to the predicated Pj, such that | dSG/dPj ⌋ ∀ Si available, is minimum.

In the above example, we assume that S3 is the minimum (and of course it is valid to recognize the physical parameters required by P4), so that the new allocation of the sensors to the predicates becomes that of Tab.II.

The implemented algorithm actually starts considering only the models of the available types of sensors and the predicate Pr to be satisfied, with possibly additional constraints (e.g., cost figures) that can be provided by the users within the sensor plugins. Then, the minimum set of sensors capable to recognize physical parameters to satisfy all the Pr is discovered and initially allocated to the most relevant predicates (in the SG sense). Based on this initial allocation, that is a pre-condition to satisfy SG, the sharing of the sensor proceeds as described in the above example. The end of the process produces a solution employing the minimum set of sensors covering all the predicates, using a quick heuristic producing a configuration that in most of the cases it is also the absolute optimum.

TABLE I. A NETWORK WITH THREE TYPES OF SENSOR.

	P1	P2	P3	P4
S1	X			
S2			X	
S3		X		

TABLE II. SHARING OF S3 BETWEEN P1 AND P4.

	P1	P2	P3	P4
S1	X			
S2			X	
S3		X		**X**

To represent how a given sensor is actually capable to capture its target events from a position (xp,yp,zp), a proper metric (3) has been defined, called *confidence*.

Confidence = 1- (Hard (xp,yp,zp,Pr))/ max Hard(x,y,z,Pr)) (3)

Where Pr is the predicate corresponding to the event, Hard (xp,yp,zp,Pr) is the hardness calculated in the candidate point for the sensor positioning and max Hard(x,y,z,Pr) is the maximum hardness within the considered environment. Note that values of confidence closer to one means that the position of the sensor is approaching the best existing in the environment to satisfy Pr, while lower values corresponds to critical points; this latter case can trigger the search for a better positioning or the increasing of the sensor set cardinality.

In the case a sensor is shared between a set of events corresponding to a group of predicate *P_set*, the confidence is calculated in the same manner, but summing the hardness of all the predicated Pr the sensor have to cover, that is:

$$\text{Confidence (event)} = 1 - \frac{\sum_{r \in P\_set} Hard(xp, yp, zp, P_r)}{\max Hard(P\_set)} \quad (4)$$

The optimization strategy can be tuned to modify this default heuristic by introducing some *taboo* conditions such as a maximum number of sharing as well as some additional figures like the cost of sensors or the requirements to achieve multiple coverage of predicates to enhance fault tolerance/reliability of the WSN response. In summary, Fig.2 depicts the pseudo-code steps of the WSN planning implemented in SWORDFISH.

```
1. Analysis of the inputs (sensing goal parsing
   and constraints processing)
2. Storing of the initial condition for the
   environment simulation
3. Profiling of the events composing the
   sensing goal (storing of the data for each
   physical parameters and point, given an
   observation window and a user defined
   sampling rate of the simulation)
4. Computation of the hardness grid for each
   predicate composing the sensing goal
5. for (numSensors=1; numSensors < maxSensors;
   numSensors++) {
        a) choice of the target predicate for
           the sensors (depending on numSensors
           and sensing goal)
        b) computation of the sensors positions
           (based on Hardness and numSensors)
        c) if (check_WSN()==OK) break}
```

Figure 2. WSN planning strategy.

IV. EXPERIMENTAL RESULTS

The goal of this section is twofold. The first is to show some practical usage of SWORDFISH, while the second is to demonstrate, through some simple but representative examples the flexibility of the approach and that, even when the complexity of the application seems to be manageable, finding the optimal solution may be not so *straightforward*.

Particular attention is paid to show the benefits of a design framework based on a formal methodology in the case of sensor sharing (to increasing the effectiveness of the WSN) and when the sensing goal is composed of a mix of physically different events. Some results are included concerning the sensitivity to some design parameters like observation time, number of sensors and position.

A. *Observation time and number of sensors,*

The goal of this example is to show the influence of the number of sensors and of the observation time onto the truth value of the sensing goal, i.e. the confidence on the capability of the WSN to correctly recognize the events.

We considered a linear model for the sensor and the sensing goal (5) corresponding to the identification of three events.

$$SG = Water(9,9,9) \text{ AND } Water(0,0,0) \text{ AND } Water(4,4,0) \quad (5)$$

The analysis result is depicted in Fig.3, showing how vary the SG when changing the observation time (time windows) and the number of sensors.

The obtained result reveals that using at least three sensors it is possible to realize a WSN capturing all of the three events disregarding the observation period.

Conversely, using less than three sensors, the time window influences the performance. With two sensors the observation time must be grater than 3 seconds: such sensors (S0, S1) will be able to recognize more than one event with the following positioning: S0=(9, 9, 4), S1=(1,2,0). In such a case, S1 can capture most of the events Water(0,0,0) and Water(4,4,0), so that S0 and S1 can cover the entire SG.

It is worth nothing that S0 is not positioned in (9,9,9), as in the case where more than three sensors are available. In fact, under this more severe "restrictions", S0 contributes to the identification of the rest of the events, though its position denotes a major value added for Water(9,9,9).

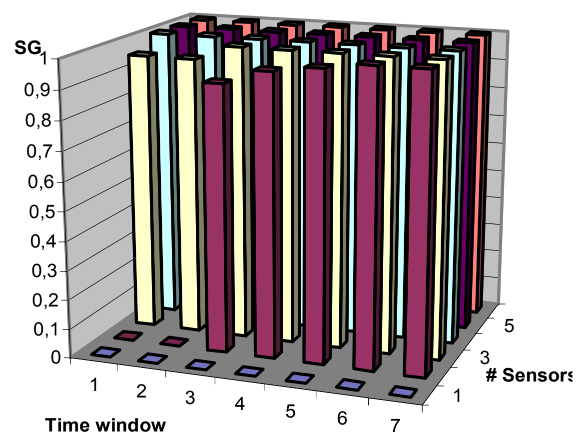

Figure 3. Sensitivity of the SG to the # sensors and time window.

B. *Influence of the sensor position*

To figure out the cross relation existing between the type of event and the position of a sensor, we considered two sensing goals (6) with a model of the sensor is still linear:

$$SG1 = Water(4,4,4); \qquad SG2 = Fire(4,4,4); \quad (6)$$

The value of SG1 has been computed considering the following space: X=[3..5], Y=[2..4] and Z=[0..3], with a time window of two seconds). The obtained data show that, to recognize a *Water* event, the sensor has to be located close to the point of interest.

Conversely, for the *Fire* event, the positioning of the sensor seems to be less important w.r.t. the previous case. This result makes sense: it possible to recognize fire events even by positioning sensors far away from the critical area. Within the entire analyzed space for the positioning of the sensors (X=[4..8], Y=[4..7] and Z=[0..8]), SG2 has been always satisfied.

C. *Sharing of sensors*

This case concerns the search for a WSN capable to recognize an event with a scarce amount of sensors. The user have to specify the max number of sensors, the min value of the SG considered acceptable and other data regarding the observation window for the sensors. We have chosen the SG (7), corresponding to the presence of water on the ground in two positions:

SG=water(0,0,0) AND water(2,2,0), with min SG=0.2 (7)

Tab.3 reports the output of SWORDFISH (sensor position) along with the confidence for each predicated. In this simple case the SG is close to one, pretty over the 0.2 threshold.

The truth value of the single predicates are similar and close to one (water(0,0,0)=water(2,2,0)=0.99999) so that the SG=0.999 is fairly acceptable. In summary, the system discovers an intermediate position for a single sensor (1,2,0) ensuring the meeting of the SG with sensor sharing. In the case our goal is modified to recognize the presence of water in two points more distant as above and with an observation window of 5 seconds, i.e.:

SG=water(0,0,0) AND water(9,9,9), with min SG=0.2 (8)

The system fails in using only one sensor and find out automatically a new WSN using two sensors, now satisfying the SG. Because of we have two sensors for two events, the suggested positioning of the sensors are obviously overlapped to the event locations (Tab.IV).

Note that Tab.IV highlights a (negligible in this case) contribute of S1 also to water(0,0,0) recognition. Such type of information can be useful to identify *Achilles' heel* of more complicated WSNs, where the amount of sensors makes hard identifying their ordering of relevance in contributing to the overall SG.

TABLE III. PLACEMENT FOR THE SET OF SENSORS.

Sens	Pos	Confidence		
		Water(0,0,0)	Water(2,2,0)	Total
S0	(1,2,0)	0.999	0.999	0.998

TABLE IV. NEW SOLUTION WITH TWO SENSORS.

Sens	Pos	Confidence		
		Water(0,0,0)	Water(9,9,9)	Total
S0	(0,0,0)	0.9999	0.0	0.999
S1	(9,9,0)	0.1	0.9999	0.999

D. *Influence of the type of event*

The WSN we are designing has the responsibility to report the presence of two different events (water and fire) having different sensing requirements.

In particular, in our modeling environment sensing the water it is harder than recognizing the fire. The SG is (9):

SG=water(0,0,0) AND fire(5,5,0), with min SG=0.4 (9)

In this case, as shown in Tab.V, the positioning of the sensor is closer (0,3,0) to the water, because of the sensing of fire is considered easier than recognizing the water itself.

TABLE V. THE TYPE OF EVENT INFLUENCES THE SENSOR PLACEMENT.

Sens	Pos	Confidence		
		Water(0,0,0)	Fire(5,5,0)	Total
S0	(0,3,0)	0.99	0.99	0.99

In more complicated scenarios, but even in this simple case, the typical design approach to place the sensors in intermediate positions disregarding the type of events to be considered, does not allows to take full advantages from the WSN intrinsic capability.

V. CONCLUDING REMARKS

The paper presented some of the design possibilities offered by the SWORDFISH framework. In particular the focus has been on the overall design methodology and on some aspects related to the sensitivity analysis of the WSN performance, with respect to some possible design choices.

The presented approach is complementary to the typical simulation-based analysis frameworks, since its emphasis is more on the system-level steps of the design, where a broad design space has to be extensively and efficiently explored, and on the formal modeling and verification of the WSN objectives.

The obtained results are promising, and some of the verification and top-level analysis and design capabilities have been addressed by considering simple but representative examples. It has been shown how it is possible to optimize the WSN while formally ensuring that the original user' goal has been fulfilled. The examples reveal that many side-effects of changing the behavior of the WSNs and sensor positioning produces strong modifications on the sensing goal that are hard to be managed by a human designer, without the support of a proper tool like SWORDFISH.

REFERENCES

[1] www.moteiv.com, www.xbow.com

[2] TOSSIM: www.tinyos.net

[3] NS-2: www.isi.edu/nsnam/ns/

[4] Avrora:research.cens.ucla.edu/projects/2006/Systsms/Avrora

[5] J-Sim: www.j-sim.org

[6] SENSE: www.ita.cs.rpi.edu/sense/

[7] OMNeT++: www.omnetpp.org

[8] VisualSense: ptolemy.eecs.berkeley.edu/visualsense

[9] SensorSim: nesl.ee.ucla.edu/projects/sensorsim/

[10] EmStar: cvs.cens.ucla.edu/emstar/

[11] OPNET: opnet.com/products/modeler/home.html

[12] J.Polley, D.Blazakis, J.McGee, D.Rusk, J.S.Baras, and M. Karir, "ATEMU: A fine-grained sensor network simulator", in Proceedings of SECON'04, First IEEE Communications Society Conference on Sensor.