

Tool-supported Requirements-based Topology Design for Wireless Sensor Networks

Stefan Lange, Jürgen Lösche, Krzysztof Piotrowski
IHP

Im Technologiepark 25
15236 Frankfurt(Oder)
Germany

Email: {langelloeschelpiotrowski}@ihp-microelectronics.com

Abstract—Planing the topology of wireless sensor networks (WSN) for a specific application is a complex task. Each application defines requirements to its WSN. Some of these requirements have to be fulfilled by the wireless technology, e.g., energy consumption and throughput, and some by the network topology, e.g., redundancy and latency. Topology makes restrictions to the wireless technology and the wireless technology makes restrictions to the network topology.

In this paper we present an algorithm to select a network topology and a wireless technology depending on application's requirements automatically.

The algorithm is part of the Sens4U approach, which aims to simplify and possibly automate the process of building WSN applications and support applications development done by non-WSN-experts.

I. INTRODUCTION

WIRELESS SENSOR NETWORK (WSN) in environmental monitoring consists of a large number of nodes with sensors and a wireless network device. These nodes are deployed over a large area, take measurements, and send the data to a sink node where the data is stored and can be analyzed. Planing of such networks is a complex and error prone task.

There are two approaches to deploy the nodes. For the first approach the sensor nodes are randomly distributed over the area, e.g., abandoned from airplane. After arriving final positions all nodes determine their geographic coordinates automatically and set up a network topology autonomously. Advantage of the approach is needlessness of a proper design.

However, the main disadvantage of this approach is the higher demand of the network nodes for systems resources. Dedicated hardware and software components are required to obtain the geographic position of the node. Furthermore, each localization method has an inherent inaccuracy which leads into divergences between the measured and the real coordinates of the node. Moreover, maintenance of network is complicated due to a lack of documentation.

The second approach contains a design phase to prepare the deployment. In this phase the positions of the network nodes are determined. During the deployment phase each sensor node is installed at its predefined location. Thus, deployment of nodes is well documented.

The toolchain introduced in the project Sens4u[1] follows the second approach. The project aims to simplify and possibly automatize the process of building a WSN application. It brings together WSN-experts and non-WSN-experts. WSN-experts can develop modules for WSN register them to the Module Pool. Non-WSN-experts are enabled to specify and build their WSN. The Sens4u toolchain transforms the WSN application specification into an implementation using the Module Pool. Thus, as a result of the project usage of WSN applications have been made available for a wide spectrum of scenarios. The proposed toolchain is given in Figure 1.

In the design flow the customer expertise into the application domain expertise is owned by the actual customer and the, at least basic, WSN expertise is owned by the integrator. The integrator role is introduced to support the customer in requirement specification. The customer explains the target application to the integrator. They identify features of the application and provides these to the planning tool as input. The planning tool generates the set of technical requirements containing the required functionalities and their required parameters. The set is forwarded to the expert system. In this component the hardware/software configuration is generated, based on the available modules in the module pool and the technical requirements. An important part of the configuration is the proposed network topology. On the one hand, several technical requirements reduce the number of allowed topologies. On the other hand, not all topologies are supported by all wireless technologies. In addition, the selection of a wireless technology results can cause side effects by including several hardware and software modules for the wireless technology. Therefore choosing a topology for the WSN application is a challenging task.

This paper focus strictly on topology selection in context of the Sens4U toolchain. All other aspects are outlined briefly only.

The remaining part of the paper is structured as follows. The following section describes the concept in detail. Section 4 represents details about the Proof-Of-Concept followed by an application example in Section 5. A section with an overview of related work follows. The paper ends with a conclusion and an outlook for future work.

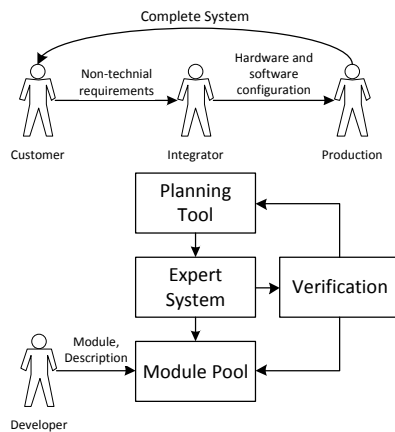


Fig. 1. The detailed tool-chain-oriented development flow and user roles.[1]

II. CONCEPT

A. Input Data

The expert system receives specifications of WSN applications from the planning tool. In this paper a network specification is defined as a tuple $S_0 = (M, s, \{R_0, R_1\}, p, t, B, R)$ consisting of

- 1) A set of measurement points M .
- 2) A sink $s \in M$.
- 3) Two reference points R_0 and R_1 with given geographically coordinates. These points are used to map the coordinates of the specification to geographically coordinates by applying cross-multiplication.
- 4) A function $p : M \cup \{R_0, R_1\} \rightarrow (x, y)$ which maps measurement points to coordinates.
- 5) A function $t : M \rightarrow T$ which maps measurement points to measurement task definitions.
- 6) A polygon B describing the outer borders of the area.
- 7) A set of requirements R . Current status is that equations $key = value$ are supported. The supported keys are given in Table I. If no requirement defines a value for a key, the default value is used for the key.

B. Architecture of the Expert System

The expert system component is the core of the Sens4U toolchain. The expert system is composed out of five subcomponents which are given in Figure 2. The figure also shows the dataflow between the subcomponents. The functionality of each subcomponent is described in the following text.

- The *Task Specification Compiler* builds from the task definition of each measurement point an application model. In addition it calculates the expected datarate during measurement operation. The result is extended to tuple $S_1 = (M, s, \{R_0, R_1\}, p, t, d, B, R)$. The function $d : M \rightarrow \mathbb{Q}$ maps each measurement point to the calculated datarate. The functionality dealing with the generated application model is not covered by this paper. For this reason the application model is not declared as an element in S_1 .

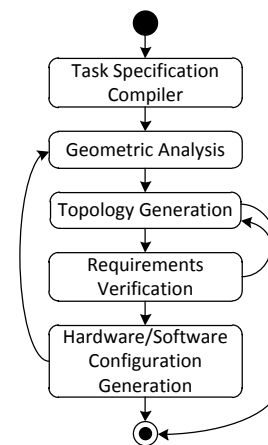


Fig. 2. Data flow in the Expert System

- During the *Geometric Analysis* the network specification is analyzed. The set M is treated as a fully connected undirected graph G_0 . The distance between its incident nodes is assigned to each edge. A graph G is created as a copy from G_0 but without all edges with a distance longer than $maxrange$. For G the values in Table II are calculated.
- *Topology Generation*: This subcomponent creates topology suggestion according to the information received from the geomtric analysis. If no topology suggestion can be made, the process exists returning an empty result. The exact behaviour of this subcomponent is described in the text later.
- The component *Requirements Verification* looks for violations of requirements caused by the topology. If a violation is found the topology suggestion is rejected and topology generation will try to create another suggestion. The utilization of defensive programming simplifies the development of new algorithm for creation of topology suggestions and defining new requirement keys. Due to this additional requirements verification the algorithms need only evaluate subsets of R .
- The *Hardware/Software Configuration Generation* tries to create hardware/software configuration for the wireless sensor nodes. It looks up the module pool for a hardware/software platform which firstly is able to implement the application model and secondly supports a wireless network technology able to form the given topology suggestion. This step can result in three different states.
 - 1) Firstly, there can be no solution. In this case the expert system terminates and returns with no solution.
 - 2) Secondly, the Expert System have found one or more possible solutions. In this case the process ends and returns the solutions to the planning tool.
 - 3) The third case means that there can be solutions with additional requirements. The topology suggestion is discarded and the additional

TABLE I
KEY/VALUE PAIRS SUPPORTED AS REQUIREMENTS

Key	Description	Default Value
<i>maxrange</i>	The maximum range of the wireless network technology.	$+\infty$
<i>rmaxdegree</i>	The maximum degree of a network node in the topology.	$+\infty$
<i>maxhops</i>	The maximum number of hops between a network data and the data sink.	$+\infty$
<i>maxdata rate</i>	The maximum throughput supported by the network technology.	$+\infty$
<i>minredundancy</i>	The minimum path redundancy of the topology.	1

TABLE II
PROPERTIES DETERMINED IN STEP 1.

Key	Description
<i>maxdistance</i>	The largest geometrical distances between two measurement points in G_0 .
<i>diameter</i>	The longest path between two nodes in G .
<i>pmaxdegree</i>	The biggest degree of a node in G .
<i>components</i>	The number of components in G .
<i>nodescount</i>	The number of vertices in G .

requirements are added to R . Then the process restarts at the geometric analysis.

C. Subcomponent Topology Generation

In this subcomponent two tasks have to be processed. The network nodes are placed and a topology suggestion for the network is created. The topology suggestion is stored as a Directed Acyclic Graph(DAG) $G_T = (V_T, A_T)$. Each vertex $v \in V_T$ represent a network node. Each directed edge $d \in A_T$ describes a suggested network link. The data flows from source to target of each edge. The root node of G_T is the data sink.

The rules how a topology suggestion is derived from a network specification are implemented in several Topology Creators. The Topology Creators are registered at process chain a priori. The Topology Creator is selected according to the requirements in M and the properties of G . For this purpose each topology creator stores a set of constraints for requirements and properties. Only under the given constraints the topology creator can create a topology suggestion.

A prolog engine is used to identify topology generators unable to fulfill given requirements. Constraints, properties, and requirements are loaded as predicates to the prolog engine. From the set of topology generators fulfilling requirements one is randomly chosen and executed. The result is sent to requirements verification component. If no topology generator is able to fulfill the requirements, the expert system terminates and return an empty result to the planning tool. The behaviour described above is given in Algorithm 1.

In the paragraphs following two Topology Generators are introduced in detail.

a) *Single-Hop Star Topology Creator*: This generator implements the most simple way to create a topology suggestion. On each measurement point a network nodes is placed and all nodes are connected directly to the sink. This algorithm works only if each node is within range of the sink.

The constraints of the Single-Hop Star Topology Generator are given in Figure 3. Lines 1 to 3 define constants. First parameter gives the name, second parameter gives the value, and third parameter gives the unit of the constant. Lines 4

to 7 define the constraints. First parameter of a constraint gives the name of the generator. The other parameters of each constraint define an equation between requirements, properties and constants. In Line 4 a constraint limiting reachable redundancy is given. A star topology does not provide any redundancy which implies a maximum value of 1 for the requirement *redundancy*. Due to, the generator does not place any repeating nodes and generates a Single-Hop topology, line 5 declares that G_1 must consists of only one component. Line 6 gives an essential but not sufficient constraint over the diameter of G . The diameter of a star is 2. Due to, the diameter of G must be less or equal to 2. However, the constraint does no check for the sink as the central node. From all measurement points data should be sent to the sink. As follows, the sink node will have a degree equal to the number of measurement points. Line 7 ensures that the requirement *rmaxdegree* allows such a topology.

b) *Multi-Hop Tree Topology*: This generator gets a minimum spanning tree from G_1 as topology suggestion. As described for the previous generator the nodes are placed at the positions of the measurement points. The constraints are given in Figure 4. This definitions differs in two points. There is no constraint on *rmaxdegree*. The generator has to check for degree of each in node in the spanning tree itself. If no valid spanning tree can be found the generator has to return an empty result. Even, there is no constraint on *diameter*. The existence of a spanning tree in a graph can not be derived from equation with the diameter of that graph.

c) *Topology Generator Selection*: The Prolog source to find topology generators, which can not fulfill the given requirements, is shown in Figure 5. Determining the suitability of a generator needs a proof of all constraints. Showing the inadequacy needs one failed proof of constraint, only. That's why *choose(T)* checks for violations and has all unusable topology generators as its result. Each valid value for T in the formular *choose(T)* gives an topology generator which cannot be used with given requirements.

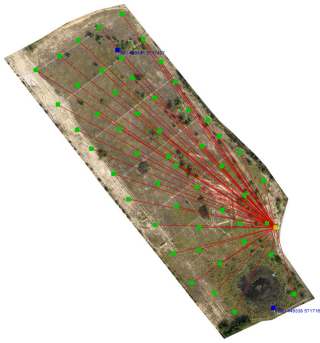


Fig. 8. A generated star topology for Hühnerwasser WSN application.

IV. RELATED WORK

In [4] the software environment POWER for planning and deploying wireless sensor networks is introduced. POWER implements an iterative process. Firstly, nodes are placed in a virtual environment. Secondly, the network in virtual environment is simulated and evaluated. If an optimal solution is found the process terminates at this point. If not, the network in the virtual environment is optimized in the third step and the process starts again. Unlike the Sens4U toolchain POWER do not enable non-WSN-experts to specify its WSN application.

The framework FLEXOR[5] defines a software architecture for wireless sensor nodes and development environment with tools dealing with WSN applications based on the FLEXOR software architecture. Like Sens4U it addresses non-WSN-experts as users of WSN applications. However, while the expert system decides about hardware and software components to use, FLEXOR supports software modules only. Furthermore, the FLEXOR-toolchain does not support topology design.

Algorithm 1 Selection of a topology generator

Require: TC is the set of all registered topology generators.

```

1: Reset Prolog Engine
2: Load  $R$  to Prolog Engine
   for each  $tc \in TC$  do
3:   Load constants of  $tc$  to Prolog Engine
4:   Load constraints of  $tc$  to Prolog Engine
5: end for
6:  $TC_0 \leftarrow$  Query Prolog Engine
7:  $TC_1 \leftarrow TC \setminus TC_0$ 
8: if  $TC_1 = \emptyset$  then
9:   return  $\emptyset$ 
10: else
11:    $G_T \leftarrow \emptyset$ 
12:   while  $G_T \neq \emptyset$  do
13:      $tc_0 \leftarrow$  first element in  $TC_1$ 
14:      $TC_1 \leftarrow TC_1 \setminus \{tc_0\}$ 
15:      $G_T \leftarrow$  Call  $tc_0$ 
16:   end while
17: end if

```

[6] defines a design flow and a user model for a component-based and composition-driven design process. The

goal is also let non-WSN-experts specify and develop WSN applications.

GENSEN[7] is a topology generator for realistic WSN deployments for the network simulator NS2[8]. Input data are not an application specification, but three parameters for node distribution. The parameters specify the distribution strategy, the number of different antenna orientations, and the number of different energy levels. GENSEN is an example for a family of topology generators for simulations.

V. CONCLUSION AND FUTURE WORK

This paper presents a solution for the difficult technical problem of topology selection that occurs in each WSN application. The approach itself and its integration into the Sens4U-toolchain are described in detail. An implementation is introduced as Proof-Of-Concept and an example calculation is given.

Several issues are still open. Sensor node placement can be prohibited by restricted zones or can become expensive by problem areas. Likewise, disturbance zones can exist where wireless communication is impossible. For these cases more sophisticated strategies for node placement have to be developed. Complex measurement tasks can be distributed in more than one node. Furthermore, generators for more effective and reliable topologies must be implemented to fulfill survivability resilience requirements.

ACKNOWLEDGMENT

This work is part of the project Sens4U (Sensorknoten für Umweltmonitoring) and was founded by the Bundesministerium für Bildung und Forschung (BMBF) under grant 03WKP26A.

REFERENCES

- [1] K. Piotrowski and S. Peter, "Sens4u: Wireless sensor network applications for environment monitoring made easy," in *SESENA*, C. Julien and K. Wehrle, Eds. IEEE, 2013. doi: 10.1109/SESENA.2013.6612264 pp. 37–42.
- [2] GEXF Working Group, "Gexf 1.2draft primer," Mar 2012.
- [3] H. Boley, "The ruleml family of web rule languages," in *Principles and Practice of Semantic Web Reasoning*, ser. Lecture Notes in Computer Science, J. Alferes, J. Bailey, W. May, and U. Schwertel, Eds., vol. 4187. Springer Berlin Heidelberg, 2006. doi: 10.1007/11853107_1 pp. 1–17.
- [4] J. Li, Y. Bai, H. Ji, J. Ma, Y. Tian, and D. Qian, "Power: Planning and deployment platform for wireless sensor networks," in *Grid and Cooperative Computing Workshops, 2006. GCCW '06. Fifth International Conference on*, 2006. doi: 10.1109/GCCW.2006.73 pp. 432–436.
- [5] A. Forster, K. Garg, D. Puccinelli, and S. Giordano, "Flexor: User friendly wireless sensor network development and deployment," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, 2012. doi: 10.1109/WoWMoM.2012.6263698 pp. 1–9.
- [6] S. Peter and P. Langendorfer, "Tool-supported methodology for component-based design of wireless sensor network applications," in *Computer Software and Applications Conference Workshops (COMP-SACW), 2012 IEEE 36th Annual*, 2012. doi: 10.1109/COMP-SACW.2012.98 pp. 526–531.
- [7] T. Camilo, J. S. Silva, A. Rodrigues, and F. Boavida, "Gensen: A topology generator for real wireless sensor networks deployment," in *Proceedings of the 5th IFIP WG 10.2 International Conference on Software Technologies for Embedded and Ubiquitous Systems*, ser. SEUS'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 436–445.
- [8] K. Fall and K. Varadhan, Eds., *The ns Manual*. The VINT Project, 2011.