Model-Driven Development of WSN Applications

A.R. Paulon, A.A. Fröhlich, L.B. Becker
Prog. Pós-Grad. em Eng. Controle e Automação (PGEAS)
Federal University of Santa Catarina,
Florianópolis, SC, Brazil
andre rp, lbecker@das.ufsc.br, guto@lisha.ufsc.br

F.P. Basso
COPPE – PESC - Federal University of Rio de Janeiro,
Rio de Janeiro, RJ – Brazil
fabiopbasso@cos.ufrj.br

EXTENDED ABSTRACT

Wireless Sensor Networks (WSNs) can be seen as distributed systems composed of tiny embedded devices, each equipped with a processing unit, a wireless communication interface, and one or more sensing and/or actuating unities[1]. It allows data-monitoring in a variety of potentially harsh environments. The most common uses of WSN are continuous sensing, event detection, location sensing, and local control of actuators.

From an application point of view, what really matters is the data collected from the WSN. This data must be forwarded from the sensing nodes to a kind of server for storage and further processing. This requires at least three different programs: 1) Sensing and data forwarding: this program runs in the embedded sensor nodes and cares for sensor-data collection and forwarding; 2) Data gathering and flushing: this program runs in a special sensor node named sink. It states for collecting the data sent by the sensor nodes and flushing them into a server or permanent storage media 3) Data storage and processing: this program runs in a server (typically not an embedded device) that receives the data collected by the sink node. It aims for data storage and further processing.

This paper presents a solution that increases the abstraction level of WSN application modeling. This allows designers to focus in the application, without needing to cope with low level details. The proposed approach aims to be applied in a Model-Driven Development (MDD) perspective[2]. The proposal includes the WiSeN Profile, created to easy the specification of WSN applications.

The rest of this paper is organized as follows. Section 2 motivates our proposal in respect to the specification of high level models for the development of WSN applications. Section 3 and 4 present the proposed profile and the model to represent WSN requirements. Finally, Section 5 concludes the paper.

WSN requirements are typically specified in structural UML diagrams, such as class diagrams. Class diagrams have classes and relationships used in a particular application. This means that each WSN application is represented in a different class diagram. Moreover, this domain owns mutually exclusive information such as technologies used to implement and to deploy a WSN system considering a set of hardware and software details (e.g. OS and APIs that map to a different source-code implementation). However, despite UML behaviour diagrams as sequence allow to represent the common behaviour among nodes, it does not allow one to represent mutually exclusive requirements. Accordingly, another abstraction must be used to specify such requirements, discussed in the next section.

In Software Engineering, requirement variabilities and communalities are commonly specified with the Feature Model (FM)[3][4]. The FM is composed by features (representing all requirements: functional, architectural, technological, mixed, etc.) from a particular technology domain. It uses relationships to configure different platform/software requirements[5]. Such possibilities allow the definition of technology variants and may be used to automate mappings between target platforms for WSN.

In the WSN domain, FM offers a holistic view of the network services for data collection and control. It represents a decision model, meaning that if the state of the application changes or if there is a modification in the application requirements at any time, it is just a matter of publishing the new desirable behavior by selecting the proper features in the domain model, as illustrated in highlighted features shown in Figure 1. This allow the application to change the system behavior directly, according with it needs for a specific WSN application.

Figure 1 contains features and relationships for a WSN domain. The overall model consists of features related to Application, Network, Programming Language, Hardware and Communication in a WSN. The model was created relating the features according to variants that we have found in WSN domain. Thus, the model becomes more intuitive and the concepts may be better reused between features, thereby promoting an applicable model.

Figure 1: Example of the Proposed Feature Model using WSN concepts

After the features of the system have been identified, a class diagram is created to be aggregate execution values for
the application’s requirements. This class diagram, describes the general structure including the components involved in sensing and data acquisition. It have the main elements of a WSN application, which has to be applied together with the right stereotypes to be effective. To this propose, the WiSeN profile, presented in next section, can be used to detail values of the system.

When applied to classes, the WiSeN stereotypes correspond to a characterization of its behavior, which indicate how the class will be employed in the application. Stereotype tags that are relevant in the context of the runtime application are mapped to classes. The code generators uses these values for attributes initialization or even for guide its transformations.

Under the offered scenarios, it is possible to configure the generation of any desired application. Nevertheless, transformers are implemented aiming the proposed architecture, that brings limitations to this kind of high-level development. The situations that are not provided by the models do not have transformers, precluding the transformation to code.

On the other hand, assuming that the chosen OS have the desired functionalities, if new concepts are aggregate to models, it is possible to implement new transformers that will result in the target application code.

The Wireless Sensor Network (WiSeN) Profile is proposed in this work to support the needs of WSN designers. WiSeN Profile can be seen as extension of the MARTE Profile. In this sense, several studies were investigated to collect the fundamental requirements of a WSN.

Motivated by the shortcomings from the related works and intending to aggregate the concepts commonly used in WSN application, this paper presents an extension to the MARTE profile. Thereby, the proposed methodology follows Selic proposal (see[6]) and applies the patterns proposed by Sadilek in[7]. The goal for using the structure already established by MARTE instead of creating a completely new profile comes from the fact that MARTE contains solid concepts related with embedded system.

Summarizing our ideas, in this paper we showed that a Feature Model and the proposed WiSen Profile can be used to transform high level models in source code. As a complement, the proposed profile is compliant with the standard MARTE profile and has been built considering current standards expressed in[8]. We consider the profile as an open proposal, subject to future refinements and extensions to address particular issues in the WSN domains (e.g., power supply control).

In this sense, we concluded that a significant set of the WSN features was covered and different WSN applications can be generated. As a natural follow-up for this work, new possible mappings to the additional WSN concepts which were not explored in this paper will be considered. Future work includes the implementation of new transformations and integrate with control systems. systems.

REFERENCES