ISSN 1884-0760

GRACE TECHNICAL REPORTS

Meta-Models for Wireless Sensor Network Applications: Data, Group, and Node Views

Ryo Shimizu Kenji Tei
Yoshiaki Fukazawa Shinichi Honiden

GRACE-TR 2012–01' February 2012

CENTER FOR GLOBAL RESEARCH IN
ADVANCED SOFTWARE SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF INFORMATICS
2-1-2 HITOTSUBASHI, CHIYODA-KU, TOKYO, JAPAN

WWW page: http://grace-center.jp/

The GRACE technical reports are published as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Meta-Models for Wireless Sensor Network Applications: Data, Group, and Node Views

Ryo Shimizu[1] Kenji Tei[2]
Yoshiaki Fukazawa[1] Shinichi Honiden[2,3]

[1] Waseda University, Tokyo, Japan
kiyo07@fuka.info.waseda.ac.jp, fukazawa@waseda.jp
[2] National Institute of Informatics, Tokyo, Japan
{tei, honiden}@nii.ac.jp
[3] The University of Tokyo, Tokyo, Japan

February 16th, 2012

Abstract

Applying model-driven development(MDD) to the wireless sensor network (WSN) domain is a promising way because MDD deals with the models from at the abstract level to at the concrete level. In this context, a developer can separately model the application logic of the WSN applications at the abstract level and a communication method and task assignment at the concrete level. However, existing studies applying the MDD to the WSN application only define the meta-model at single abstraction level. In this report we define the meta-models at three abstraction levels. We expect to achieve the MDD process for WSN applications using the models at multiple abstraction levels with our meta-models.

1 Introduction

A wireless sensor network (WSN) is a network consisting of small devices, called sensor nodes, with a wireless communication unit, CPU, RAM, multiple sensors, and a battery. They also have resource constraints, e.g. energy and reliability. Due to these constraints, the WSN application developer has to optimize the quality of the sensor data, e.g. the data sensing span and data loss rate. Because the data quality heavily depends on the execution environment [6], the developer tunes the data quality by making design decisions such as a communication method and task assignment. To optimize performance, applications are needed to be executed on the environment at early stage of development. In this way, the WSN application development

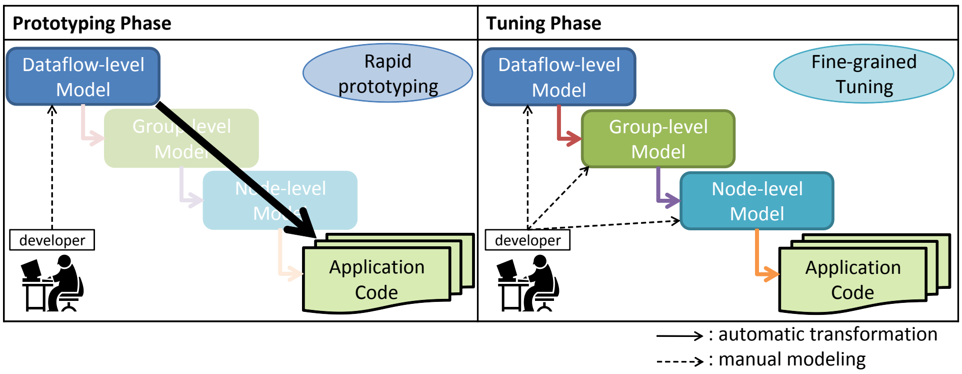

Figure 1: The overview of the MDD process of the WSN applications.

requires the following: a rapid prototyping that is to develop the prototype in a low-cost way, and a fine-grained tuning that is to tune the data quality by the communication methods and task assignments.

The developer usually describes WSN applications by using domain specific language (DSL) to abstract the details of WSNs. A lot of DSLs are proposed and they can be classified to three levels, dataflow-, group-, and node-level, based on a scope of programs [4]. The node-level DSLs express the instructions for the individual node. The group-level DSLs express the macro-behavior of the node group. The dataflow-level DSLs only express data sampling and processing. The developer often adopts the node-level DSLs, because these DSLs can finely tune the data quality in detail. However, it is difficult for non-WSN experts to use the node-level DSLs because it requires the knowledge of the WSN. Hence, dataflow- and group-level DSLs are suitable for non-WSN experts. These DSLs can express the WSN application simply whereas they cannot express in-depth designs for tuning the data quality. Therefore, the developer cannot satisfy two requirements for the WSN application development by using single DSL.

Model-driven development (MDD) is a promising solution to satisfy these requirements at the same time. In the MDD process, the developer can describe the applications as an abstract model, then iteratively refine the model into concrete ones and the corresponding code. In this context, the developer can describe the WSN application at the dataflow-level at first. If the tuning of the data quality is needed, the developer should design the application at the group- and node-level as well as the dataflow-level. We illustrated the overview of this process in Figure 1.

However, existing work using the MDD to the WSN application development only focused on the single abstraction level. The MDD framework named Baobab [1] has provided the meta-model for the WSN application and code generation system. Their meta-model represented node-level DSLs. Losilla et al. have defined a meta-model for group-level DSLs, and the code

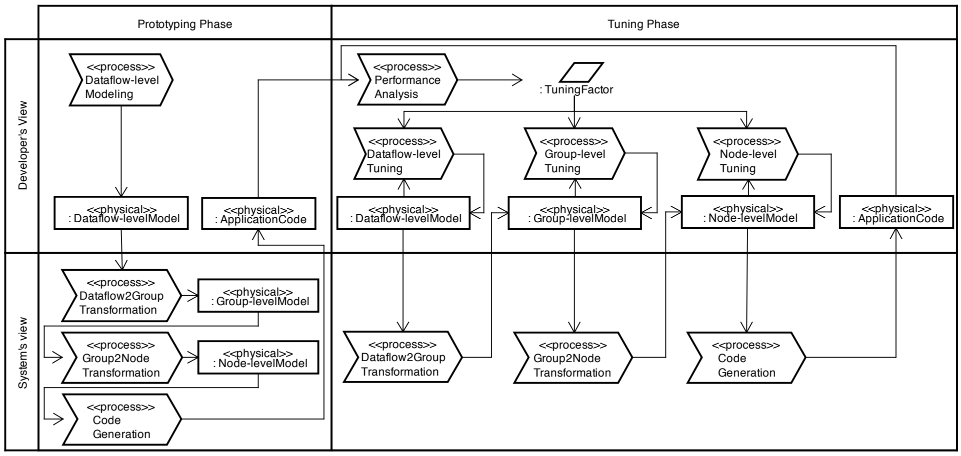

Figure 2: The development process using multiple abstraction levels.

is automatically generated from the model [3]. Meanwhile, our previous work [5] proposed the development process using multiple abstraction levels, illustrated in Figure 2. We defined the meta-models at three abstraction levels and transformation rules between the models described based on our meta-models. However, the meta-models in [5] is not enough to describe the real-world WSN applications.

In this report, we describe the definition of meta-models that are the modified version of meta-models in [5]. We defined three meta-models based on the existing DSLs and the classification of the DSLs. The dataflow-level DSML expresses a network-independent dataflow, that is application logic of the WSN application. The group-level DSML can be described a configuration of the leader-member type node group. The node-level DSML contains the role which represents a group of tasks and the node in the WSN, and expresses a role assignment to nodes. These DSMLs are designed to describe the most typical WSN applications that samples and sends at fixed interval [2].

2 The Dataflow-level Meta-Model

We defined the dataflow-level meta-model to expresses a network-independent dataflow, that is an application logic of the WSN application because data flow is the basic application logic of WSN applications. The dataflow-level model consists of a data source, intermediate processing point, data sink and relations between them.

Figure 3 shows the meta-model at dataflow-level. The data source (`DataSource` in the meta-model) needs information of a data type to sense, duration of data sampling and transmission, and location of the data source.

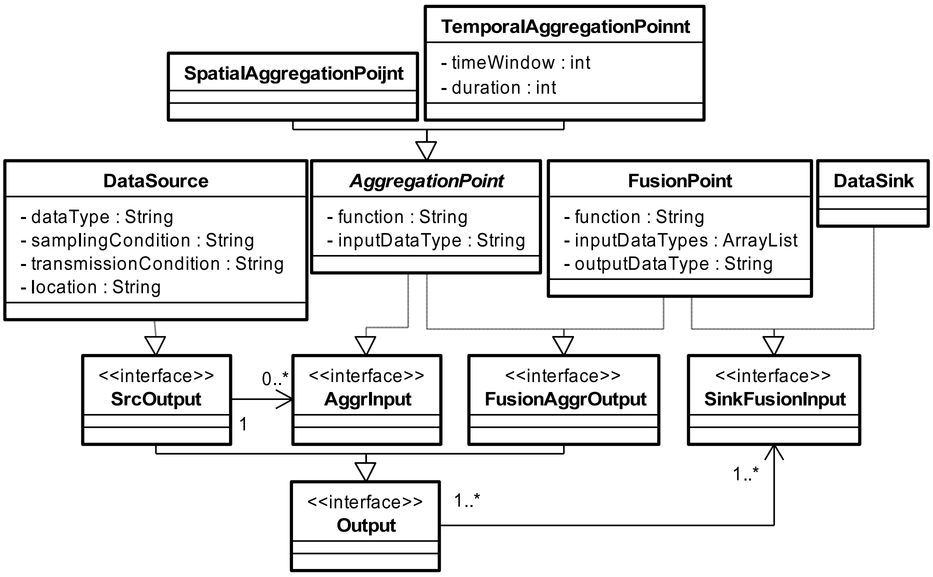

Figure 3: The dataflow-level meta-model.

The data processing point has two processing types: aggregation (`AggregationPoint`) dealing with a single data type and fusion (`FusionPoint`) dealing with multiple data types into a single data type. The aggregation also has two types: a temporal aggregation (`TemporalAggregationPoint`) and spatial aggregation (`SpatialAggregationPoint`). The temporal aggregation deals with the aggregation of time-series data and the spatial aggregation deals with the aggregation of data that is spatially distributed. They need attributes for the operation and data type dealing with. The temporal aggregation additionally needs parameters for aggregation, a width of time window and duration. Treating data is finally collected to the data sink (`DataSink`). The developer describe dataflow-level model by deciding the relations between aforementioned objects and attributes in these objects.

3 The Group-level Meta-Model

We defined the group-level meta-model to express the configuration and behavior of node groups. The group-level model is dependent on the network and describable about handling a group of nodes as a single entity, thus the developer can model the application in more simple way than model the behavior of the individual node. Each group in the model is the leader-members type group.

Figure 4 shows the definition of the group-level meta-model. The group (`Group`) in our meta-model consists of the leader, member, and commu-

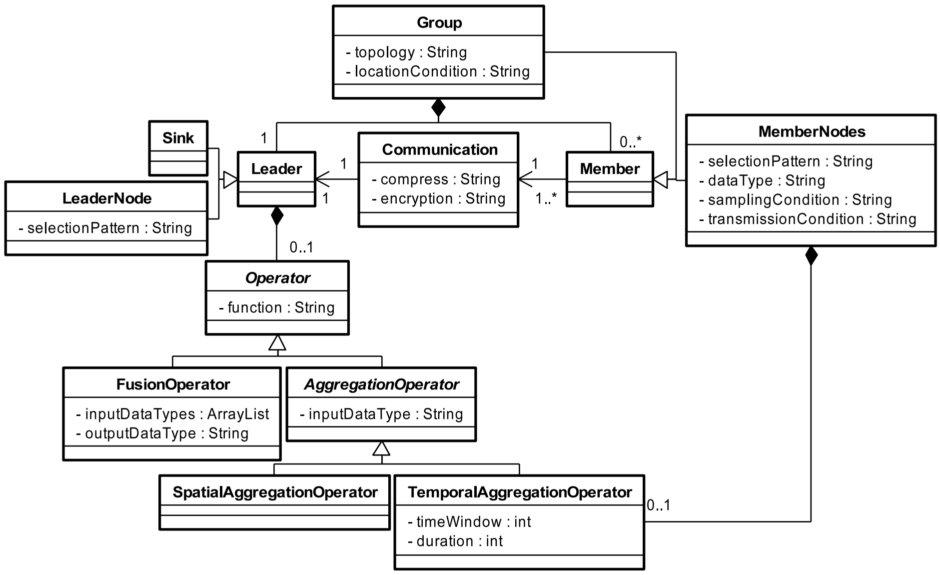

Figure 4: The group-level meta-model.

nications from the members to the leader (`Communication`), and contains information of the network topology and location of the group. In our meta-model, the leader (`Leader`) is the generalization of a data sink (`Sink`) and leader node (`LeaderNode`) and the member (`Member`) is the generalization of a member group and member nodes (`MemberNodes`). The member nodes yield a sensor data and the nodes need information on the data sampling, and duration of data sampling and transmission. The developer can model the in-/out-network aggregation by assigning a data processing task (represented by `Operator`) to the sink/leader node. The data processing includes a data fusion (`FusionOperator`), temporal aggregation (`TempralAggregationOperator`), and spatial aggregation (`SpatialAggregationOperator`) same as at dataflow-level. The leader node and member nodes have a deployment condition that is the conditional expression of the task assignments to the nodes in WSN. A data compression and encryption are modeled as the attributes of the communication.

4 The Node-level Meta-Model

The node-level meta-model is defined to express a behavior and configuration of each node because WSN applications are finally executed on the individual node. The combination of tasks such as data sampling and the task assignment represent the behavior.

In node-level model, a role that represents the group of task (`Role`) has relations to sensor nodes (`Node`) and these relations represent the task assign-

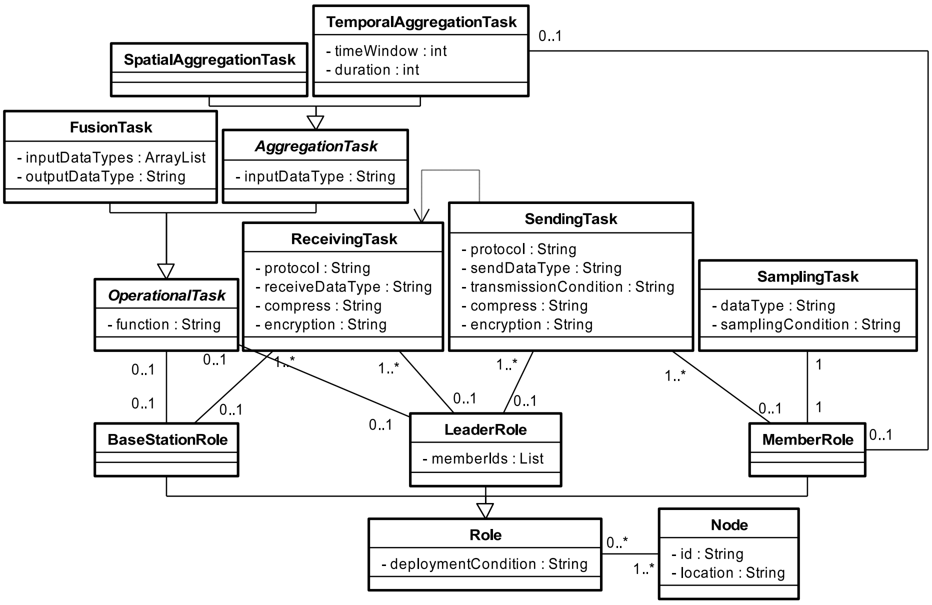

Figure 5: The node-level meta-model.

ments. We define three types of role, the role as a group member (`MemberRole`), group leader (`LeaderRole`), and base station (`BaseStationRole`). The member role is responsible for the task of data sampling (`SamplingTask`) and transmission (`SendingTask`). The leader role contains a relaying task, data receiving (`ReceivingTask`) and transmission, and a task of in-network data processing (`OperationalTask`). The base station role collects the transmitted data, and then processes the data. Same as at the dataflow- and group-level, the data processing contains the fusion (`FusionTask`), temporal aggregation (`TemporalAggregationTask`), spatial aggregation (`SpatialAggregationTask`). Each task has attributes to define the specific behavior and configuration, e.g. the sampling task has the attributes of the data type and sampling period. As for the communication between nodes, a routing protocol, data compression, encryption, and data type they dealing with are modeled as the attributes in both the receiving and sending side. Figure 5 illustrates the definition of the node-level meta-model.

5 Conclusion

In this report, we described the definition of three meta-models. The MDD process of the WSN application using multiple abstraction levels is a promising way to achieve both the rapid prototyping and fine-grained tuning. However, existing work only focus on the single abstraction level. We define the dataflow-, group-, and node-level meta-model based on the existing DSLs for the WSN applications, and describe the details of each meta-model.

In the future, we will evaluate the description capabilities of three meta-models with the real-world WSN applications. With the transformation rules between the models described base on our meta-models, we will also evaluate the applicability of the development process we proposed in [5].

References

[1] Bahar Akbal-Delibas, Pruet Boonma, and Junichi Suzuki. Extensible and precise modeling for wireless sensor networks. In Jianhua Yang, Athula Ginige, Heinrich C. Mayr, Ralf-D. Kutsche, Wil Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, and Clemens Szyperski, editors, *Information Systems: Modeling, Development, and Integration*, volume 20 of *Lecture Notes in Business Information Processing*, pages 551–562. Springer Berlin Heidelberg, 2009.

[2] Lan S. Bai, Robert P. Dick, and Peter A. Dinda. Archetype-based designf sensor network programming for application experts, not just programming experts. In *Proceedings of the 2010 Fourth International Conference on Sensor Technologies and Applications*, IPSN '09, pages 323–328, Washington, DC, USA, 2009. IEEE Computer Society.

[3] Fernando Losilla, Cristina Vicente-Chicote, Bárbara Álvarez, Andrés Ibora, and Pedro Sánchez. Wireless sensor network application development: An architecture-centric mde approach. In Flavio Oquendo, editor, *Software Architecture*, volume 4758 of *Lecture Notes in Computer Science*, pages 179–194. Springer Berlin / Heidelberg, 2007.

[4] Luca Mottola and Gian Pietro Picco. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Comput. Surv.*, 43(3):19:1–19:51, April 2011.

[5] Ryo Shimizu, Kenji Tei, Yoshiaki Fukazawa, and Shinichi Honiden. Model driven development for rapid prototyping and optimization of wireless sensor network applications. In *Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications*, SESENA '11, pages 31–36, New York, NY, USA, 2011. ACM.

[6] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 1–13, New York, NY, USA, 2003. ACM.