RESTIoT: A Model-based Approach for building RESTful Web Services in IoT Systems

Darwin Alulema[1,2], Javier Criado[2], Luis Iribarne[2]

[1] Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador
[2] Applied Computing Group, University of Almería, Spain

Abstract. In today's Internet of Things (IoT) world, we face many challenges, such as having an architecture that improves the integration of data coming from several IoT devices and supports an efficient building of systems able to monitor, send and process data in real time. In this sense, we propose a model based on Model-Driven Engineering (MDE) for automatic RESTful web services building in IoT systems. This paper presents a work in progress for the development of a graphic editor and a transformation engine proposed to simplify the creation of RESTful web services related to the operation of the IoT hardware nodes (RESTIoT). This approach semi-automatically generates Java code (which uses Spring Boot technology) for web services and Arduino code for IoT nodes controllers.

Keywords: Model-Driven Engineering (MDE), Internet of Things (IoT), Web Services, Domain-Specific Language (DSL)

1 Introduction

In recent years, we have witnessed how the use of Internet has become popular and the applications using it have grown in complexity to the point of talking about a fourth industrial revolution [8], whose impact was envisioned by Kevin Ashton, who introduced the term IoT in 1999 [5]. Ever since, the term IoT has been used in many areas to refer to the ability to generate, send, store and process information without human intervention. This development has allowed the appearance of new application scenarios such as *Smart Health*, *Smart Cities* and *Smart Homes*, among others. In addition, since the price of sensors is now much lower than before and wireless connections have greatly improved, the relevance of the IoT has continued growing [6]. However, the IoT is currently affecting the functioning of the Internet at all its levels, from addressing to the processing of information [1].

In the IoT, a trend is getting objects to communicate with each other and collaborate with their neighbors to achieve a common goal. Due to the increase of applications based on IoT, this communication is growing and counts a large number of different objects, many of them interacting directly with their physical environment [7]. Therefore, ease of interaction is a key requirement for IoT

systems, it must be present in architectures that allow access to such devices [5] (*e.g.*, a Service-Oriented Architecture, SOA).

To improve the application development process, the availability of a language with a concrete set of concepts specific to the IoT domain may be of particular interest. Such a language should allow developers to define the system so that it can be used for automatic or semi-automatic generation of code specific to each platform. To this end, our approach combines the advantages of Model-Driven Engineering (MDE) and microservices to semiautomate the generation of IoT applications. From a Domain-Specific Language (DSL), we derive all necessary tools to achieve an IoT system consisting of a set of RESTful web services and a database that conforms to the definition of the target devices.

There are previous studies where MDE was used to create RESTful web services [4, 9–11]. Other research focuses on RESTful web services applied to IoT [1–3, 5]. However, there are only a few projects that incorporate these technologies [6]. For this reason, our approach introduces a web service that is simpler than those exhibited in previous studies (complex events, deployment in multiple platforms, security), but with the feature of being easily incorporated to platforms commonly used in the IoT (*e.g.*, SmartPhone, wearables, web and DTV).

2 The RESTIoT solution

The proposed DSL (see Figure 1) consists of `IoTNode` elements representing an *ad-hoc* node hardware abstraction for the application to be developed. These nodes are represented as a collection of devices that may be `Sensor`, `Actuator`, or `Controller`, and that can be linked to a `Port` of a controller. The approach also makes it possible to define a set of attributes associated with the Web `Services` and `Databases`. Meanwhile, each service has a set of `Operations` to access the resources created in the databases. The implementation of these

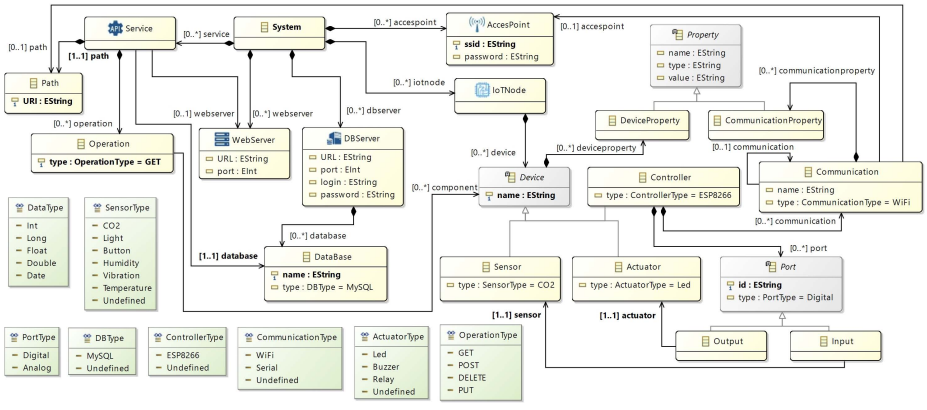

Fig. 1. DSL for the construction of RESTful web services in IoT systems.

operations is built from `Properties` (in the traditional manner: name, type and value) that can have type `Devices` (*i.e.*, `Sensor`, `Actuator` and `Controller`) and properties that are associated with the service. These services are then accessed by the associated node or another client through a provided `Path`.

The defined DSL allows the creation of software tools that can be later used by developers of IoT applications. In our case, we have developed two tools according to the semantics of this DSL. The first consists of a **graphical editor** created with Sirius, in which a tool palette is provided to model IoT scenarios (see Figure 2). The second software tool defined compliant with the DSL is a **transformation engine**, implemented in Acceleo, which allows model-to-text (M2T) transformations to generate code for both the client and the server sides. For the client node, Arduino code is generated, and for the service, Java code is created in the framework of Spring Boot technology, because it reduces the amount of code necessary for the configuration, which is better suited to Acceleo. In Listing 1, we show a piece of code that is generated from the tool for a web service that allows querying node information stored in a database.

3 Conclusions and Future Work

This paper has proposed a solution for the IoT systems creation. To achieve this goal, we have developed two tools using MDE techniques: a graphical editor and

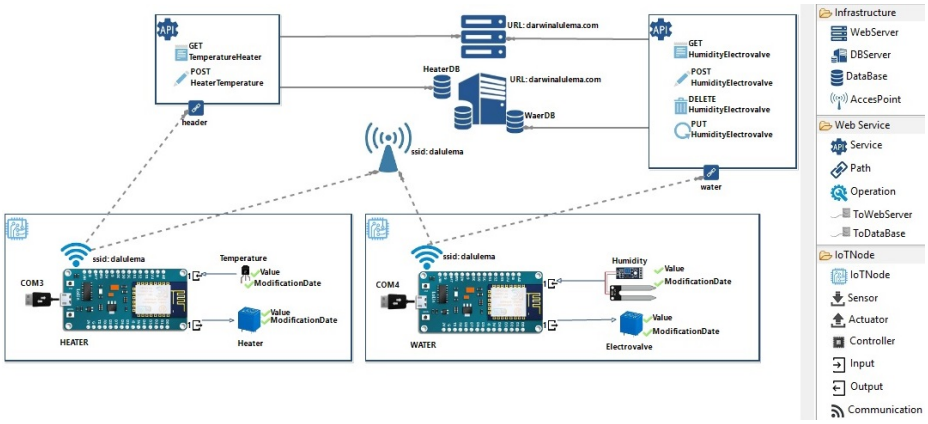

Fig. 2. Partial view of the graphic editor to model IoT scenarios.

```
1  @GetMapping("/devices/{id}")
2  public ResponseEntity<IoT> getIoTById(
3      @PathVariable(value = "id") Long id) {
4      return new ResponseEntity(
5      iotRepositorio.findById(id),HttpStatus.OK);
6  }
```

Listing 1. Fragment of sample code of a generated web service.

an M2T transformation engine. The approach allows developers to define the system using a graphical tool that allows them to describe the structure of IoT nodes and their links with web services. As a result, we get a semi-automatic process for Java code generation using Spring Boot technology for RESTful web services and Arduino code for the deployment of IoT nodes.

For future work, the following lines of research have been identified: a) extending the DSL to include security issues in web services, b) improve the interoperability between the IoT nodes incorporating publish-subscribe standards, and c) conduct performance testing of the services and usability testing of the tools.

Acknowledgments: Work funded by TIN2017-83964-R and CEIMAR / CeiA3.

References

1. Alonso, Á., Fernández, F. Marco, L. and Salvachúa, J.: IAACaaS: IoT Application-Scoped Access Control as a Service. Future Internet. 9(4):64. (2017)
2. Boubeta-Puig, J., Ortiz, G., and Medina-Bulo, I.: Knowledge-Based Systems MEdit4CEP: A Model-Driven Solution for Real-Time Decision Making in SOA 2.0. Knowledge-Based Syst. 89:97–112. (2015)
3. De Melo Silva, C., Cerqueira, H., De Sousa Júnior, R., Buiati, F. and García, L.: Design and Evaluation of a Services Interface for the Internet of Things. Wirel. Pers. Commun. 91(4):1711–1748. (2016)
4. Ed-douibi, H., Cánovas, J., Gómez, A., Tisi, M., and Cabot, J.: EMF-REST: Generation of RESTful APIs from Models. Annual ACM Symposium on Applied Computing. pp. 1446–1453. (2016)
5. Garcia-de-Prado, A., Ortiz, G., and Boubeta-Puig, J.: COLLECT: COLLaborativE ConText-Aware Service Oriented Architecture for Intelligent Decision-Making in the Internet of Things. Expert Syst. Appl. 85:231–48. (2017)
6. González, C., Pelayo, C., Pascual, J. and Cueva-Fernandez, G.: Midgar: Generation of Heterogeneous Objects Interconnecting Applications. A DSL Proposal for IoT Scenarios. Comput. Netw. 64:143–158. (2014)
7. Patel, P., and Cassou, D.: Enabling High-Level Application Development for the Internet of Things. J. Syst. Softw. 103:62–84. (2015)
8. Qafmolla, X. and Nguyen V.: Automation of Web Services Development Using Model Driven Techniques. Int. Conf. Comp. & Aut. Eng. pp. 190–194. (2010)
9. Tavares, N. and Vale, S.: A Model Driven Approach for the Development of Semantic RESTful Web Services. Proceedings of International Conference on Information Integration and Web-based Applications and Services. pp. 290–300. (2013)
10. Terzić, B., Dimitrieski, V., Kordić, S., Milosavljević, G., and Luković, I.: Development and Evaluation of MicroBuilder: A Model-Driven Tool for the Specification of REST Microservice Software Architectures. Enterp. Inf. Syst. 12(8-9):1–24. (2018)
11. Zolotas, C., Diamantopoulos, T., Chatzidimitriou, K. and Symeonidis, A.: From Requirements to Source Code: A Model-Driven Engineering Approach for RESTful Web Services. Automat. Softw. Eng. 24(4):791–838. (2017)