WHITE PAPER

Tools Work Group

oneDM & OCF Core Framework

USING SDF DEFINED OBJECTS ON OCF

July 15, 2020

Introduction

Industry collaboration is key to the success of the Internet of Things (IoT). One of the issues is having a common data models across the industry. To solve that issue, Open Connectivity Foundation has worked under liaison with Bluetooth SIG, OMA SpecWorks, Zigbee Alliance on One Data Model. The work has progressed in such a way that the conducted work is now made public and the continuation of the work will be conducted in public.

Semantic Definition Format (SDF)

OneDM has creating a new definition format to convey data models with the end goal of standardizing submitted data models defined by the various organizations so that a single model for a desired feature or purpose can be selected. The developed definition format is called Semantic Definition Format (SDF). SDF has fulfilled requirements from the participating organizations. SDF has fulfilled requirements from the participating organizations, for example:

- Format should be machine readable
- Format should support syntax validation
- Format should be suitable for code generation
- Easy to read, easy to understand
- Syntax should be rich enough to replace the current data modeling mechanisms by the participating organizations.
- Not tied to transport protocols.

SDF is rich enough to convey typical data modeling constructs as:

- Objects
- Actions
- Events
- Read/write indication on parameters
- Strong typed data for parameters
- Referencing to definitions, so that a high-level reuse can be created

This means that the modeling is not tied to transport constructs/methodologies such as SOAP or Restful. This allows data to be defined as needed for the function that needs to be designed. For example, a light can be designed as a state variable to turn the light on and off, but also as an action can be defined to invert the current state of the light. The latter can be used to make multi-way switching possible.

SDF is represented in JSON, which facilitates the creation of tools. For validation of the data expressed in a data model, SDF uses components from the JSON-based schema language defined by json-schema.org, reducing the amount of work to create validation tools.

For example, JSON is widely supported with node.js and python. Libraries exist and examples of how to use them are available on the internet. Furthermore, JSON documents can be checked against a JSON schema. Also using JSON as syntax reduces the amount of work to create a validation tool. Having the schema validation in place for the data models is helping to create data models that are compliant with SDF.

Learn more about SDF here.

To validate that SDF is rich enough in features, OCF and other participating organizations have converted a subset of their data models into SDF format. SDF data models are published under the BSD-3 License, making the data models available for consumption by everyone. The OCF contributed models in SDF format are automatically converted by tooling. The OCF tooling is available here. The setup of the tooling is scripted, so that the tooling is easy to install and use. The conversion uses the already machine readable and public available OCF datamodels as input and can be automatically translated into SDF by a single command. The SDF data models are made publicly available on the OneDM playground here. The data models on the playground are submitted by the contributing organizations for evaluation. Furthermore, the playground has CI tooling in place to automatically check the schema compliance of the submitted models.

SDF used for code generation

To validate the usability of the models in SDF, OCF has taken an additional step by converting SDF models to Open API specification 2.0 (OAS2.0). OAS2.0 is the data model language that OCF is using. OCF has tooling available that converts the OAS2.0 models into code. This is a two -step process. The first step is selecting which OAS2.0 based resources should be incorporated into the device. Any resource type (“rt”) identified in the input by a unique URL can be used in the implementation. Additionally, optional properties can be removed from the resource definition. The result of this step is a single file that includes all the necessary information of all the resources that makes up the device. The 2nd step is to use the assembled file as input for code generation. The code generation tool is template based. By using templates, the tool can support different computer languages, operation systems and stacks. The default set up is to use the combination of C code/Linux-windows/IoTivity-Lite stack. The generated code is

fully functional and can be used as starting point to create OCF devices. More information about the tool chain can be found at IoTivity Getting Started page. This tutorial describes how to get started with the IoTivity-Lite code base and uses a binary switch resource as example resource using the code generation tool set.

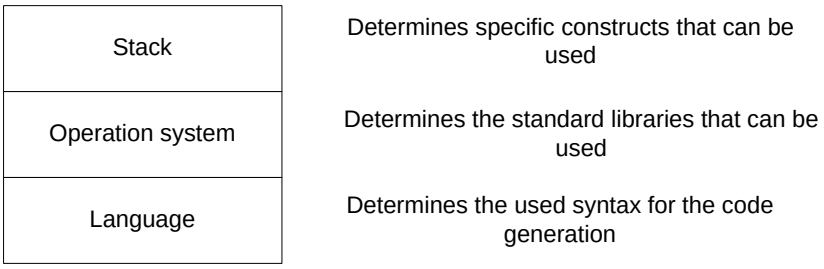

Figure 1 Dependencies for building a code generator

The OAS2.0 files generated from SDF can also be used with the code generation tool set. Tooling for conversion are available here. For example, the analog input file contributed by IPSO can be used as input. The process includes the conversion of an SDF object to a resource description in OAS2.0 format. Since the converted model is not an OCF approved model the rt type is not prefixed with “oic.r.” the conversion tooling is using the OCF vendor extensions mechanism for data models, e.g. using the “x.<vendordomain>” prefix. The conversion steps are described in the figure blow.

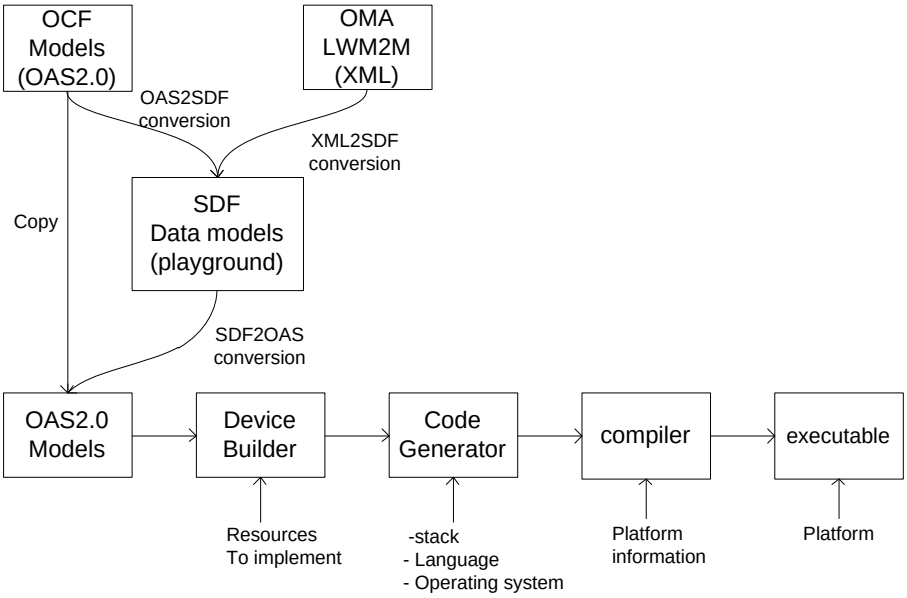

Figure 2 tool chain for data model conversion and code generation

The conversion tools are based mappings between SDF and the other data model definitions, the map between the formats are described here. This approach allows that the OCF core framework can transport a mixed set of data models defined by multiple organizations.

References

SDF:

https://github.com/one-data-model/SDF

SDF playground:

https://github.com/one-data-model/playground/

Open Connectivity Foundation (OCF)

ISO/IEC 30118

Standardized rt values:

https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#rt-link-target-att-value

IoTivity-Lite getting started:

https://iotivity.github.io/IoTivity-Getting-Started/

Conversion tooling:

https://github.com/openconnectivityfoundation/SDFtooling

SDF mapping to other data model definitions:

https://onedm.org/terminology/