Workgroup: ASDF Working Group
Internet-Draft: draft-laari-asdf-relations-00
Published: 3 June 2022
Intended Status: Informational
Expires: 5 December 2022
Authors: P. Laari
Ericsson

Extended relation information for Semantic Definition Format (SDF)

Abstract

The Semantic Definition Format (SDF) base specification defines set of basic information elements that can be used for describing a large share of the existing data models from different ecosystems. While these data models are typically very simple, such as basic sensors definitions, more complex models, and in particular bigger systems, benefit from ability to describe additional information on how different definitions relate to each other. This document specifies an extension to SDF for describing complex relationships and additional information about them.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the A Semantic Definition Format for Data and Interactions of Things Working Group mailing list (asdf@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/asdf/.

Source for this draft and an issue tracker can be found at https://github.com/plaari/draft-laari-asdf-relations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 5 December 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The Semantic Definition Format (SDF) [SDF] is a format for domain experts to use in the creation and maintenance of data and interaction models in the Internet of Things. The SDF specification defines a generic data model that can be used as a meta model when converting between other data models, such as IPSO Smart Objects or Digital Twins Definition Language (DTDL) [DTDL]. SDF model defines a set of affordances, describing the interfaces for the Object. These can be mapped to corresponding affordances in other data models.

The base specification defines ways to represent parent-child relations between two definitions. However, there is a need to describe also more complex relations to support arbitrary connections between definitions and also referring to definitions outside of the SDF models. These could be, for example, defining possible location of a device inside a room, how a device is controlled by another device, or physical topology between devices. This enables defining more complex systems using SDF models.

The basic parent-child relations between SDF Objects and Things can be defined by including a definition of a child in the definition of the parent. This covers a large share of simple data models defining, e.g., simple sensors, or more complex devices containing a set of sensors. On the other hand, SDF can be used also to describe even more complex entities, such as buildings with rooms and other related objects inside a building. When we extend the SDF usage, the simple relation is often not enough, but more complex relations may be needed to describe the connections between the definitions. These relations can be for example physical (e.g., an object is inside another object), functional (e.g., an object can control another object), or semantic (e.g., an object is similar to a term defined in another ontology).

This document extends the base SDF specification by adding a new key word to describe also other relations between physical or logical objects. This new key word is needed to describe, without loss of information, models from ecosystems that are using complex relation information in their definitions.

NOTE: This extension is now defined based on the Relationships feature in the DTDL specification. There may be other kind of definitions for relationships in other data models that must be taken into account and this specification may need to be extended to cover also those requirements.

2. Terminology

This specification uses the terminology specified in [SDF], in particular "Class Name Keyword", "Object", and "Affordance".

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL** NOT", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. SDF Relation Extension

In this section we define a new SDF Class Name Keyword, sdfRelation, that can be used to describe complex relations. The definitions are

on class-level, i.e., the sdfRelation keyword does not give any instance specific information about the relation, but defines the potential relations between definitions.

3.1. Namespaces

The SDF namespace block can be used to provide CURIE prefixes for external ontologies for use with sdfRelation extension. For example, in case of SAREF (Smart Applications REFerence ontology) ontology extension for buildings [saref4bldg], we can use the following namespace definition:

```
{
  "namespace": {
    "saref": "https://saref.etsi.org/saref4bldg/v1.1.2/"
  }
}
```

3.2. Qualities of sdfRelation

In this section, the qualities of the sdfRelation are defined. These qualities are used to define the potential type of the connection between the definitions and to which definition the connection can be made.

Quality	Type	Required	Description
relType	string/ IRI?	no	What kind of relationship these definitions have
target	string	no	Target definition for the relation
description	string	no	Description of the relationship
maxItems	integer	no	Maximum number of instances of the target types
minItems	integer	no	Minimum number of instances of the target types
property	object	no	Additional properties for this relation
writable	boolean	no	Is the target writable or not

Table 1

3.2.1. relType

The relType quality describes what kind of relationship this definition has to the target definition. This can use different ontologies, such as SAREF from ETSI. The used ontology **MUST** be defined in the namespace block to give a short name for the ontology IRI.

For example the "relType" field could define the relationship to be saref:isControlledByDevice, when the SAREF ontology is used with CURIE prefix "saref" defined in the namespace block for the full IRI https://saref.etsi.org/saref4bldg/v1.1.2/. The defined purpose for the relation is a functional relationship between the two definitions.

3.2.2. target

The "target" field defines to which definition or ontology term this definition with sdfRelation has a relation to. This can be e.g. #/sdfObject/room, when the target object is defined in the same SDF model. This may also be left undefined, and in that case the relation may be any other object (Note: This is from DTDL (check), does it make sense in SDF context?)

The target does not have to be another SDF object, but it can be also a reference to another ontology. For example, we may have a Temperature sensor, which relation to SAREF temperature sensor is defined and it is the same as this one.

```
"namespace": {
  "exont": "https://example.com/relationOntology",
  "saref": "https://saref.etsi.org/core/v3.1.1/"
},
sdfObject: {
  "temperature": {
    "description": "Example temperature object",
    "sdfProperty": {
      ...
    },
    "sdfRelation": {
      "sameAs": {
        "relType": "exont:same-as",
        "target": "saref:TemperatureSensor"
      }
    }
  }
  ...
```

3.2.3. description

The description of the relationship. For SDF version 1.1, the description is a string. (For future SDF versions this description can be localizable, allowing different languages in the description.)

3.2.4. maxItems

Maximum number of instances of the target definition that can be related to this definition. If not specified, the number of instances is not limited.

3.2.5. minItems

The minimum number of instances of the target definition that must exist for this definition. If defined, this value **MUST** be between zero and maxItems. Default: 0.

3.2.6. property

Object with key-value pairs that describe additional properties for this relationship. Details TBD.

3.2.7. writable

Is the information of the relation writable, i.e., can be changed. Default: false.

3.3. Example relation description

In the following example, we have a definition for first-object which located next to second-object:

```
 "namespace": {
   "exont": "https://example.com/relationOntology"
 },
 sdfObject: {
   "first-object": {
     "description": "Example object",
     "sdfProperty": {
       ...
     },
     "sdfRelation": {
       "next": {
         "relType": "exont:next-to",
         "target": "#/sdfObject/second-object"
       }
     }
   },
   "second-object": {
     "description": "Example object, next to the first object",
     "sdfProperty": {
       ...
     },
     "sdfRelation": {
       "next": {
         "relType": "exont:next-to",
         "target": "#/sdfObject/first-object"
       }
     }
   }
 }
```

4. SDF DTDL mapping

This section (to be removed) shows mapping between SDF and DTDL qualities for relations.

Quality (SDF)	Quality (DTDL)	Description	Required
sdfRelation	@type	In DTDL, this is "Relationship", this is the objects sdfRelation entity	yes
"name-of-relation"	name	In SDF, this is the entity name	yes
relType	@id	DTDL: The ID of the relationship description	no
writable	writable	Boolean, is this relation writable or not	no
target	target		no

Quality (SDF)	Quality (DTDL)	Description	Required
		An Interface ID, in SDF the target definition	
$comment	comment	This is for model authors in DTDL	no
description	description	DTDL: localizable description for display	no
	displayName	DTDL: localizable name for display	no
property	properties	A set of Properties that define relationship-specific state	no
maxItems	maxMultiplicity	max nof target instances	no
minItems	minMultiplicity	min nof target instances	no

Table 2

5. Security Considerations

TODO Security

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[SDF] Koster, M. and C. Bormann, "Semantic Definition Format (SDF) for Data and Interactions of Things", Work in Progress, Internet-Draft, draft-ietf-asdf-sdf-11, 28 February 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-asdf-sdf-11>.

7.2. Informative References

[DTDL] "Digital Twins Definition Language (DTDL) v2", 10 February 2022, <https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDL/v2/dtdlv2.md>.

[saref4bldg]
Poveda-Villalón, M. and R. Garcia-Castro, "SAREF extension for building", 5 June 2020, <https://saref.etsi.org/saref4bldg>.

Acknowledgments

The author wants to thank Ari Keranen, Mikko Saarisalo, and Christer Holmberg for their feedback and comments.

Author's Address

Petri Laari
Ericsson

Email: petri.laari@ericsson.com