ASDF H. Lee, Ed.
Internet-Draft J. Hong
Intended status: Informational ETRI
Expires: 27 March 2026 23 September 2025

# Semantic Definition Format (SDF) modeling for Digital Twin
draft-ietf-asdf-digital-twin-01

## Abstract

This memo specifies SDF modeling for a digital twin, i.e. a digital twin system, and its Things. An SDF is a format that is used to create and maintain data and interaction, and to represent the various kinds of data that is exchanged for these interactions. The SDF format can be used to model the characteristics, behavior and interactions of Things, i.e. physical objects, in a digital twin that contain Things as components.

## Status of This Memo



## Copyright Notice

Table of Contents



## 1. Introduction

A digital twin is defined as a digital representation of an object of interest and may require different capabilities, for example, synchronization and real-time support, according to the specific domain of application. [Y.4600]. Digital twin help organizations improve important functional objectives, including real-time control, off-line analytics, and predictive maintenance, by modeling and simulating objects in the real world. Therefore, it is important for a digital twin to represent as much real-world information about the object as possible when digitally representing the object.

Nowadays, digital twin technologies are applied in various domains including manufacturing, energy, medical, farm, transportation, etc. And a common format is needed to represent the objects in the domains as digital twins. SDF [I-D.ietf-asdf-sdf] can be used for modeling objects as digital twins.

This document specifies the modeling and guidance on how to use SDF to represent objects as digital twins.

## 2. Terminology

This specification uses the terminology specified in [I-D.ietf-asdf-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 structure for digital twin

This section describes SDF structure with the new Class Name Keyword, sdfContext, to represent a thing or an object as a digital twin. The architecture of a digital twin based on the SDF model is illustrated in Figure 1, , following the guidelines of [ISO23247-3].

The Physical Layer comprises affordance and non-affordance objects. From the real-world objects, only those deemed relevant are selected for representation as digital twins. The Digital Twin Layer is structured into three sublayers: the Device Communication Sublayer, the Digital Twin Sublayer, and the Application Sublayer. The Device Communication Sublayer is responsible for monitoring and collecting data from both affordance and non-affordance objects. This sublayer provides the necessary data to synchronize the physical objects with their digital twin counterparts. The Digital Twin Sublayer ensures synchronization between the affordance and non-affordance objects and their respective digital twins using the data provided by the Device Communication Sublayer. The Application Sublayer presents the synchronized values of the digital twins to users, facilitating informed decision-making.

```
        +---------------------------------------------+ - - - - - - - - - - -
        |            Application Sublayer              |
        | +----------+ +------+ +--------+ +--------+ |
        | |  Human   | | HMI  | |  Apps  | |  Peers | |
        | +----------+ +------+ +--------+ +--------+ |
        +---------------------------------------------+
        |           Digital Twin Sublayer             |
        | +----------+ +-------------+ +------------+ |
        | | Operation| | Application | | Resource   | |
        | |    and   | |     and     | | access and | |
        | |management| |   service   | |interchange | |
        | +----------+ +-------------+ +------------+ |
        | +-----------------------------------------+ |  Digital twin Layer
        | |     Digital representation of objects   | |
        | |   +-------------+   +----------------+  | |
        | |   |  Affordance |   | Non-affordance |  | |
        | |   |   objects   |   |    objects     |  | |
        | |   +-------------+   +----------------+  | |
        | +-----------------------------------------+ |
        +---------------------------------------------+
        |         Device Communication Sublayer       |
        |     +-------------+   +----------------+    |
        |     |     Data    |   |     Object     |    |
        |     | collection  |   |     control    |    |
        |     +-------------+   +----------------+    |
        +---------------------------------------------+ - - - - - - - - - - -
        |     +-------------+   +----------------+    |
        |     |  Affordance |   |  sdfContext    |    |
        |     |   objects   |   |    objects     |    |     Physical Layer
        |     +-------------+   +----------------+    |
        +---------------------------------------------+ - - - - - - - - - - -
```

Figure 1: Basic Architecture of digital twin

## 4.  Motivation and design rationale

The document is based on the underlying structure defined in [I-D.ietf-asdf-sdf], which which standardizes the semantic definition format (SDF) for representing IoT affordance.  This specification provides a strong basis for representing individual devices and their features (sdfProperty, sdfAction, sdfEvent, etc.), but additional mechanisms are needed to address the unique requirements of digital twin modeling.

Digital twin systems defined in [ISO23247-3] often have to describe virtual representations of various physical assets, including metadata, identity, contextual relationships, historical data, as well as device interfaces.

### 4.1. Introduction of sdfContext

A new SDF keyword sdfContext described in [I-D.draft-ietf-asdf-sdf-nonaffordance] is introduced to represent non-functional or metadata elements that describe a device or component without implying direct interaction:

* Identifier (e.g., UUID, URN)

* Location (e.g. site, zone, GPS tag)

* Owner (e.g., representative, ,anufacturer)

These field can appear in both sdfObject and sdfThing contexts and follow the same structural pattern as sdfData and is designed for scalability.

### 4.2. Digital Twin-Centric Modeling within sdfThing

To support hierarchical representations (e.g., a boat composed of heater, GPS, and battery subsystems), this document encourages use of sdfThing to aggregate related sdfObject components, along with metadata.

The mapping of digital twin attributes to sdf qualities are shown in Table 1.

| Attribute | Recommended Mapping | Description |
|---|---|---|
| Identifier | sdfContext | Globally unique digital twin ID (e.g., URN) |
| Characteristic | sdfProperty or sdfData | General description or domain properties |
| Schedule | sdfEvent or sdfData | Time-based actions, availability, or maintenance |
| Status | sdfAction or sdfProperty | Actual or calculated operating conditions |
| Location | sdfContext | Physical or logical location information |
| Report | sdfData | Measurement summaries, analytics, or logs |
| owner | sdfContext | Organization or entity responsible for the digital twin |
| Relationship | sdfRelation | Inter-object/inter-twin relationships |

Table 1: Digital twin modeling within sdfThing

# 5. Examples

## 5.1. Boat modeling

The example of boat007 Figure 2illustrates how a Digital Twin representation can be constructed for a heater component (heater1) installed on a specific vessel (boat007). The Digital Twin is modeled using the sdfThing structure, which references the heater object defined in the sdfObject section.

```
{
  "sdfThing": {
    "boat007": {
      "label": "Boat #007 with a heater",
      "description": "Contains heaters, fans, and so on"
      "sdfProperty": {
        "status": {
```

```
                  "type": "boolean",
                  "description": "Indicates if the boat is powered"
                }
              },
              "sdfObject": {
                "heater": {
                  "description": "A heater ",
                  "identityManifest": {
                    "manufacturer": "HeaterTech Inc.",
                    "model": "HEATER-2025-V1",
                    "firmwareVersion": "1.4.3",
                    "dateOfManufacture": "2025-04-20T09:00:00Z",
                    "certifications": [
                      { "scheme": "KS", "certId": "KS123456", "region": "KR" }
                    ]
                  },
                  "contextSnapshot": {
                      "thingId": "heater:unit5689",
                      "timestamp": "2025-05-23T10:20:00Z",
                      "installationInfo": {
                          "room": "kitchen",
                          "floor": 1,
                          "mountType": "freestanding",
                          "installationDate": "2025-06-01"
                        },
                        "usageProfile": {
                          "type": "residential",
                          "powerCircuit": "230V@60Hz",
                          "energyRating": "A++"
                        },
                        "location": {
                          "lat": 35.1796,
                          "lon": 129.0756
                        }
                  },
                  "sdfProperty": {
                    "status": {
                      "type": "boolean"
                      "description":"Indicates if the heater is powered"
                      },
                      "temperature": {
                         "sdfRef": "#/sdfProproperty/temperature",
                         "maximum": 8
                      }
                  },
                  "contextPatch": {
                    "thingId": "heater:unit5689",
                    "timestamp": "2025-06-20T09:00:00Z",
```

```
                             "location": { "lat": "35.2988", "lon": "129.2547", "alt": "0
.0" },
                             "installationInfo": { "floor": 1, "mountType": "wall" }
                           }
                         },
                         "thermostat": {
                           "maintenanceSchedule": {
                            "timestamp": "2025-05-20T10:00:00Z"
                            "description": "Last maintained date"
                            }
                         }
                     }
                   }
                 }
               }
```

Figure 2: An example of SDF mapping for digital twin

5.2. Relationship modeling

sdfRelation, defined in [I-D.draft-laari-asdf-relations], is a structure for specifying logical or physical relationships between objects within an SDF model. If conventional sdfThing, sdfObject, and sdfProperty focus on defining the properties of individual digital twins, sdfRelation is a means of expressing interactions and structural links between them. Since these relationships go beyond a single digital twin definition, they must be managed in a separate structure, where sdfRelation is used. The sdfRelation keyword allows describing complex relationships beyond just the parent-child hierarchy. These relationships can include:

* Physical relations (e.g., "inside", "next to")

* Functional relations (e.g., "controls", "is controlled by")

* Semantic relations (e.g., "similar to", "same as")

The sdfRelation definition can include the following fields as defined in [I-D.draft-laari-asdf-relations]:

* relType: Specifies the type of relationship that can an external ontologies (e.g., SAREF[saref4bldg]) can refer to.

* target: Points to the SDF object or an external ontology term that is the target of the relationship.

* description: Provides a detailed textual explanation of the relationship.

* label: A short human-readable label for the relationship.

* property: Additional properties describing the relationship context.

* $comment: Optional properties including implementers notes.

```
                    {
                      "info": {
                        "title": "Example lightbult and thermostat using sdfRelation"
                      },
                      "sdfObject": {
                        "lightbulb": {
                          "description": "A smart lightbulb",
                          "sdfProperty": {
                            "adjacent-node": {
                              "type": "object",
                              "sdfType": "link"
                            }
                          },
                          "sdfRelation": {
                            "sameRoomAsThermostat": {
                              "relType": "saref:isLocatedIn",
                              "target": "#/sdfObject/thermostat",
                              "description": "This lightbulb is located in the same room
 as the thermostat.",
                              "label": "Located together"
                            }
                          }
                        },
                        "thermostat": {
                          "description": "A thermostat is in the same room as the lightb
ulb",
                          "sdfProperty": {
                            "adjacent-node": {
                              "type": "object",
                              "sdfType": "link"
                            }
                          }
                        }
                       "sdfProtocolMap": {
                          "description": "Protocol between the lightbulb and thermostat"
,
                          "ble": {
                            "services": [
                               {
                                "serviceID": "361c9c4f-22d7-4a1e-824b-8b61045a566a"
                               }
                              ],
                              "cached": false,
                              "cacheIdlePurge": 3600,
```

```
                    "unit": "Second",
                    "autoUpdate": true,
                    "bonding": "default"
                }
              }
            }
          }
```


Figure 3: An example of sdfRelation

## 6. Requirements for digital twin

A digital twin is a partial representation of sdfThing or sdfObject that contains attributes such as sdfProperty, sdfAction and sdfEvent[ISO23247-1]. By representing sdfThing as a digital twin, crucial events that require appropriate action can be quickly detected and controlled. The requirements defined in [ISO23247-1] are applied to represent sdfThings and sdfObjects as digital twins.

* Identification: sdfThings and sdfObjects should contain data that uniquely identify them as digital twins.

* Data acquisition: data related to sdfThing and sdfObject, such as sdfProperty, sdfEvent, and sdfAction, should be collected from IP and non-IP devices.

* Data analysis: collected data needs to be analyzed to understand the state of sdfThing and sdfObject.

* Accuracy: The sdfThings and sdfObjects should be represented as digital twins with appropriate levels of detail and accuracy, depending on the application.

* Synchronization: sdfThings and sdfObjects should be synchronized with the digital twin at intervals appropriate to the requirements of each application. Newly added or deleted sdfThings and sdfObjects should be recognized and reflected in the digital twin.

## 7. Procedure for digital twin implementation

### 7.1. Overview

It is essential to define a standardized implementation procedure to ensure interoperability, scalability, and effective lifecycle management across digital twin systems. This section outlines a step-by-step approach aligned with the Semantic Definition Format (SDF) model and its architecture, enabling consistent modeling, integration, and operation of digital twins in IoT environments. A general principles for representing an sdfThing as a digital twin within a specific domain is outlined as follows:

* defining a purpose for expressing the observable object, as known as a physical asset or an object of interest, as a digital twin in the domain

* organizing data based on the roles of the observable object in the domain

* configuring the observable object into the digital twin based on the data for the purposes

* interworking with a digital twin of each of other domains in which the observable object performs a different role

* synchronizing the observable object and the digital twin

### 7.2. Procedure

The procedure of digitally twinning the space and the objects contained in it is described.

* Identifying and scoping physical assets: The first step is to clearly identify the physical assets that will be represented as digital twins. This step includes assigning a globally unique identifier, such as a URN or UUID, and determining the extent of modeling. It also involves deciding whether the unique identifier will cover the entire system or focus on a specific subsystem or component. Although all assets in space can be represented by digital twins, it is cost-effective to select assets for implementation purposes and configure them as digital twins.

* Defining a digital twin: A detailed digital twin should be defined using SDF structures, including sdfThing and sdfObject. This step requires specifying affordances such as sdfProperty, sdfAction, and sdfEvent, as well as non-affordance metadata like location, owner, and other descriptive elements through sdfContext.

* Metadata and contextualization: This step adds metadata that enriches the context of the digital twin, such as geographic location, ownership details, manufacturing information, and feature summaries. It can also support advanced analytics and management, including contextual attributes such as production schedules or maintenance periods.

* Binding interfaces and communications: Digital twins are bound to real-world communication interfaces and protocols such as MQTT, CoAP, and HTTP. This allows affordance of SDF models to interact with real-world data sources, APIs, and physical assets in a smooth and reliable manner.

* Verification and compliance: Once an asset is defined and bound as a digital twin, it should be validated against syntax and semantic rules using tools such as JSON schema validators or CDDL definitions. Compliance with specific SDF profiles or domain-specific standards must also be verified to ensure interoperability.

* Deployment and registration: After verification, the digital twins are deployed in a digital twin registry, edge system, or cloud infrastructure. This step involves registering the model with the discovery service for integration and use by other systems or stakeholders.

* Runtime monitoring and updating: During operations, digital twins need to continuously monitor real data and update their status accordingly. Properties updates, event processing, and partial updates using contextPatch messages should be supported for efficient and lightweight synchronization.

* Lifecycle and governance management: The life cycle of the digital twin is managed through version tracking, audit logs, and compliance documents. This step ensures safe and transparent governance and enables proper disposal and deregistration when assets are no longer available.

## 8. Security Considerations

Only authorized users should have the authority to manage digital twins, sdfThings and sdfObjects. Also, Secure communication and metadata integrity are essential when implementing digital twins. All context messages, including contextPatch and identityManifest, must have mechanisms such as authentication and authorization applied.

## 9. IANA Considerations

This document has no IANA actions.

## 10. References

### 10.1. Normative References

[I-D.draft-ietf-asdf-sdf-nonaffordance]
Hong, J. and H. Lee, "Semantic Definition Format (SDF) Extension for Non-Affordance Information", Work in Progress, Internet-Draft, I-D.draft-ietf-asdf-sdf-nonaffordance-01, 21 September 2025, <https://datatracker.ietf.org/doc/html/I-D.draft-ietf-asdf-sdf-nonaffordance-01>.

[I-D.draft-laari-asdf-relations]
Laari, P., "Extended relation information for Semantic Definition Format (SDF)", Work in Progress, Internet-Draft, I-D.draft-laari-asdf-relations-04, 28 January 2025, <https://datatracker.ietf.org/doc/html/I-D.draft-laari-asdf-relations-04>.

[I-D.ietf-asdf-sdf]
Koster, M., Bormann, C., and A. Ker辰nen, "Semantic Definition Format (SDF) for Data and Interactions of Things", Work in Progress, Internet-Draft, draft-ietf-asdf-sdf-24, 27 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-asdf-sdf-24>.

[ISO23247-1]
"Automation systems and integration Digital twin framework for manufacturing - Part 1: Overview and general principles, ISO 23247-1.", October 2021, <https://www.iso.org/standard/75066.html>.

[ISO23247-3]
"Automation systems and integration Digital twin framework for manufacturing - Part 3: Digital representation of manufacturing elements, ISO 23247-3.", October 2021, <https://www.iso.org/standard/78744.html>.

[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>.

[Y.4600] Union, I. T., ""Recommendation ITU-T Y.4600 (2022), Requirements and capabilities of a digital twin system for smart cities.", August 2022.

## 10.2. Informative References

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

## Acknowledgements

This specification is based on work by the One Data Model group.

## Contributors

Joo-Sang Youn
DONG-EUI University
176 Eomgwangno Busan_jin_gu
Busan
47340
South Korea
Phone: +82 51 890 1993
Email: joosang.youn@gmail.com

Yong-Geun Hong
Daejeon University
62 Daehak-ro, Dong-gu
Daejeon
34520
South Korea
Phone: +82 42 280 4841
Email: yonggeun.hong@gmail.com

## Authors’ Addresses

Hyunjeong Lee (editor)
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu
Daejeon
34129
South Korea
Phone: +82 42 860 1213
Email: hjlee294@etri.re.kr

Jungha Hong
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu

Daejeon
34129
South Korea
Phone: +82 42 860 0926
Email: jhong@etri.re.kr