

ASDF
IETF 110
2021-03-11

Chairs: Michael Richardson, Niklas Widell

Note Well

- You will be recorded
- Be nice, and be professional
- The IPR guidelines of the IETF apply:
see <http://ietf.org/ipr> for details.

Repo: <https://github.com/ietf-wg-asdf/asdf-working-group-notes>

Notes: <https://codimd.ietf.org/notes-ietf-110-asdf>

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (<https://www.ietf.org/contact/ombudsteam/>) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- [BCP 9](#) (Internet Standards Process)
- [BCP 25](#) (Working Group processes)
- [BCP 25](#) (Anti-Harassment Procedures)
- [BCP 54](#) (Code of Conduct)
- [BCP 78](#) (Copyright)
- [BCP 79](#) (Patents, Participation)
- <https://www.ietf.org/privacy-policy/>(Privacy Policy)



Agenda

- Introduction, meeting logistics, agenda bashing
 - Note takers, jabber scribe
- ASDF Outreach
- Hackathon Results
- ASDF status update (chairs)
 - SDF 1.1 Consensus Call
- Future planned virtual Interims
- SDF.next
 - P/A/E; scaling
 - “Mapping”
 - SDF Compact
- AOB

Administrative

- CodiMD for notes <https://codimd.ietf.org/notes-ietf-110-asdf>
- Meetecho: <https://meetings.conf.meetecho.com/ietf110/?group=asdf&short=&item=1>
- jabber: <xmpp:asdf@jabber.ietf.org?join>
- All ASDF notes at:
<https://github.com/ietf-wg-asdf/asdf-working-group-notes.git>

WG procedures

- Decisions on mailing list: <https://www.ietf.org/mailman/listinfo/asdf>
- Work on Github: <https://github.com/ietf-wg-asdf>
 - Use Issue tracker for issues (new features and fixes)
- Schedule (doodle) regular virtual interims between virtual physical meetings

Status update

- **Status:** ASDF WG was chartered in October 2021
- **Chairs:** Michael Richardson and Niklas Widell
- **Progress so far:**
 - IETF 109 with Hackathon
 - Two virtual interims
 - SDF 1.1 on Call for Consensus
 - IETF 110 Hackathon last week, discussion of new features
- **Meeting plans:**
 - ASDF at 110 (this meeting)
 - Additional interims to be planned

ASDF Outreach

- Value of ASDF increases with increased adoption of SDF by organisations doing IoT data models
- Outreach so far (beyond OneDM)
 - ASDF presented to DMSE (LwM2M) group in OMA SpecWorks
 - SDF proposed to ISO/IEC JTC1 SC41 for IoT Thing modeling
 - W3C Web of Things
 - Other use cases: Electronic Data Sheets, Textile manufacturing
- **If your organisation works with IoT data models, we want to hear from you!**

IETF 110 Hackathon

- Collaboration ASDF and WISHI
- Daily one-hour synch meetings during Hackathon week (Mo-Thu)
- Participation: Seven contributors, including two newcomers, one of which a student
- Outcome:
 - Lots of exploratory discussions on new features (compact formats, event specification, extensions, mapping files, SDF-WoT alignment, ...)
 - Two tools to simplify model translation
 - Collaboration with W3C WoT

ASDF/WISHI Hackathon

- validate SDF1.1, move tools and models up to SDF1.1
 - Testing with MJK's MODBUS example (via node-red)
 - Defined fragment identifier considerations (PR #20)
 - Identified scaleMinimum/scaleMaximum problem: digital vs. physical facet of a model
 - Tangible result: draft-bormann-asdf-sdf-compact-00, SDF compact notation
- continue work on the "semantic proxy"
(connect OMA and OCF clients/servers guided by common SDF)
 - IPSO/SDF converter, IPSO mapping file example, Demo LWM2M device + WoT TD
- SDF Thingmaker (Petri Laari) — assemble sdfThings out of sdfObject models
- continue development of the "mapping file" concept for:
 - ecosystem specific mapping files (IPSO above, ZCL example in repo)
 - Protocol bindings in mapping files (+ Jan Romann's pointer reversal proposal)

SDF 1.1 – Call for Consensus

- Call for Consensus for ASDF WG to adopt draft-ietf-asdf-sdf-03 and mark as Implementation Draft SDF 1.1 started on February 23rd and concluded on March 9th.
- **No objections were raised => We now (almost) have SDF1.1**

Major changes 1.0 => 1.1

- Clarify cross-referencing/reuse, sdfRef
- Support composite data types
- Specify two kinds of choice/enumerations

Implementation Draft?

The intent of the Implementation Draft status is to identify a stable intermediate specification that implementations and toolchains can be targeted at, while work on the specification continues for additional required features.

SDF 1.1: ~~One~~ 3 more things

- Around IETF110/hackathon, three PRs were generated that should go into 1.1:
- <https://github.com/ietf-wg-asdf/SDF/pull/21> (#21): I-D nits; IANA considerations section for application/sdf+json.
- <https://github.com/ietf-wg-asdf/SDF/pull/24> (#24): make info block optional: Tools composing SDF pieces no longer have to detect and discard dummy info blocks.
- <https://github.com/ietf-wg-asdf/SDF/pull/25> (#25): scaleMin/aximum → SDF 1.0.
 - They never were fully defined, so removing these is not actually a change.
 - They were not used so far in the models in the OneDM playground.
 - While we now have a proposed definition (in #22), they actually don't work the way they should when combining model and mappings.
- → No changes to format. Merge before publishing -04.

SDF.next – going forward

- With a stable SDF1.1 Implementation Draft it is time to review what additional features and supporting functionality the SDF RFC will need
- On the agenda for this meeting
 - Clarifications of Properties, Actions & Events
 - Mapping files
 - Compact format

Planning for next virtual interims

SDF 1.1 and 1.next clarifications

Carsten Bormann, with input from Michael Koster

ASDF@IETF 110, 2021-03-11

Agenda

- (1) Properties/Actions/Events
- (2) Data items vs. physical quantities
 - type="number", scaleMinimum/Maximum, multipleOf, unit

(1) Properties, Actions, Events

- Property: $[T \Rightarrow] \blacksquare [\Rightarrow T]$
(if writable) (if readable, observable)
- Action: $T_1 \Rightarrow \blacksquare \Rightarrow T_0$
- Event: $\blacksquare \Rightarrow T$
- Property and Action are distinct, but what about Property and Event?

Events

- Very different concepts are called “Events”
- The door bell: a genuine Event
 - Difference between pressing it once or twice (Postman!)
 - Except during a short instance, the state is always the same
- A sale: a genuine Event
 - Precious: must not be lost
 - Event comes with data that are only about the event

Non-Events

(Not really)

- Temperature Sensor with Threshold Alarm:
 - Property (has a current value, can be read)
 - Notification function (alarm = observable)
- Confusion: some properties need to have history about state changes
 - Threshold excursions may be reason to populate history
 - But history currently not modeled in SDF
 - Probably worth looking at: draft-bormann-t2trg-stp

Clarifications needed: sdfProperty/observable

- *rwo* bits — *readable, writable, observable* (all default to `true`, BTW!)
- Due to CoAP observe (which implies a GET):
one might think that *observable* implies *readable*.
But that is not the way it is defined today!
- Independent bits mean that a `--o` property is possible:
observable, but not *readable* (maybe better term: *notifiable*)
 - Sensor that just reports (periodically, at threshold), but cannot be “read”
 - Pubsub-style, but not in an `sdfEvent` sense
- We can model this today!

Clarifications needed: sdfEvent

- Defined by its signature: Event: $\blacksquare \Rightarrow T$
- Not to be used where an observable property is the right description. I.e., here, T is not a copy of the state!
- Requires some attention in 1.next cycle:
 - Preciousness
 - Queueing issues, overload
- But, with care, can be used today

Clarifications needed: sdfAction

- Current model should work well
- Not much pressure-testing though
- Can't model P/A/E spawned from an sdfAction yet (cf. Section 2.3 of draft-ietf-ace-aif “dynamic”) — *dynamic affordances*
- No clarifications currently required
- But expect further work on sdfAction

(2) Data items vs. physical quantities

- SDF describes **Information Model**
- Ecosystems may have different digital representations for same information
- Potential perception:
json-schema.org-derived qualities describe **digital** side (bits)
- Common information model really should be about **physical** side (quantities)

SDF 1.1: Digital vs. Physical

The picture with scaleMin/aximum

	Digital	Physical
What is it	type="number" (I-JSON issues)	(Implied)
What is the range; fence posts included?	minimum, maximum; exclusiveMinimum, exclusiveMaximum	scaleMinimum, scaleMaximum
What is the precision/ granularity	multipleOf	—
Unit	—	unit

- Modeling has started on the digital side (copying from json-schema.org)
- But digital side may differ between ecosystems: often should be described in *mapping* only
- SDF model: should be Physical first!
 - Describe digital side only if common (converged)
 - Leave to mapping otherwise

Example

- Temperature sensor, scale -50°C to $+150^{\circ}\text{C}$
- Digital side in centiKelvins offset to Celsius, -500 to $+1500$
- (digital) minimum = -500 , maximum = $+1500$
- scaleMinimum = -50 , scaleMaximum = $+150$, unit = $^{\circ}\text{C}$ (Cel)
- Digital minimum/maximum could be supplied by the mapping info (As could be scaleMinimum and scaleMaximum)
- Which side gets the json-schema.org keywords minimum/maximum?

Survey (OneDM playground)

- No scaleMin/aximum
- minimum and maximum actually are **physical** quantities for IPSO models
- _____ insert info about other sets _____

- Conclusion: use JSO-inspired minimum and maximum for physical side
- Consequence: need to stay in I-JSON constraints, but IEEE 754 has been designed so that isn't a problem for most physical quantities

SDF 1.1: way forward

With a hint how 1.next could look like

	Digital	Physical
What is it	type="number" (I-JSON issues)	
What is the range; fence posts included?	digitalMinimum, digitalMaximum vs. scale/offset [RFC 8798]	minimum, maximum; exclusiveMinimum, exclusiveMaximum
What is the precision/ granularity	(Implicit via scaling)	multipleOf
Unit	—	unit

- JSO “minimum” and “maximum” are well-entrenched
- Keep that on the **physical** side
- 1.next: Introduce **new** scaling qualities on the **digital** side, e.g.:
 - digitalMinimum/digitalMaximum (scaleMin/aximum turned around)
 - Maybe even scale/offset (if there are no clear ranges)

(What we don't want to do:) Clarifications needed

- (See table)
- How do digital and physical qualities combine?
 - Unfounded hope: simple “augmentation model” of mapping can work
- Physical first: start with `scaleMinimum`, `scaleMaximum`
- Leave `minimum`, `maximum`, `multipleOf` to mapping
 - Unless the digital side is common (converged)

(What we don't want to do:) Fix 1.1!

- type="number"
 - Is now understood to describe both physical and digital side
 - does not imply I-JSON constraints on the physical side
- type="integer"
 - Probably should do same (i.e., only for physically countable things)
- Clarification: multipleOf is on the digital side, so multipleOf=1 implies
 - Integer on digital side
 - No implication on physical side

(What we don't want to do:) Fix 1.1, continued

- scaleMinimum, scaleMaximum: wrong terms
 - physicalMinimum, physicalMaximum
quantityMinimum, quantityMaximum ??? (or stick with old?)
 - Only specify in the information model if there is a physical reason (e.g., normalized 0..1 or 0..100, or non-negative such as K)
- exclusiveMinimum and exclusiveMaximum are on the digital side, too
 - Currently allows both JSO 4 (Boolean) and JSO 7 (number)
 - Should decide now (playground is JSO 4 because of OCF JSO)

ASDF@IETF 110

Mapping Files

Michael Koster

March 11, 2021

Mapping Files - Problem

- SDF definitions are meant to be protocol neutral and easily hosted over diverse protocols and storage schemes
- Best design practice for e.g. OneDM is to leave as much of the definition to implementation time while allowing for syntactic interoperability within an ecosystem and semantic interoperability across ecosystems
- Mapping files are a way to separate the protocol-agnostic part of the model from implementations

Mapping Files

- External information model elements/augmentation
- Add qualities to further constrain a data type
 - minimum, maximum, multipleOf
 - abstract number to typed number
 - reserve space, indicate position in a packed representation
- Select a subset of enumeration values
- Set value defaults, set values to constants
- Add new qualities (ID codes, custom settings)
- Add protocol binding settings
- Over-ride certain elements (defaults...)

Processing model

- Mapping information per SDF element
 - sdfThing, sdfObject, sdfEvent, sdfAction, sdfProperty, sdfData, sdfEnum
- Mapping elements are processed as if the definitions appear in the source file
- Mapping extensions should be defined and validated by extension schemas
- Mapping extension vocabularies should be able to use existing SDF facilities, e.g. sdfRef

Mapping file format

- Top level JSON object with selectors as keys and SDF definition and extension elements as values
 - Selectors use JSON Pointer with the same reference model as used in SDF proper, e.g. sdfRequired
 - Including the SDF namespace facility

Example, override number constraint and add custom qualities to a definition in the playground:

```
{  
  "pg: /#/sdfObject/Voltage/sdfProperty/Sensor_Value": {  
    "ipso_id": "5700",  
    "type": "float",  
  },  
}
```

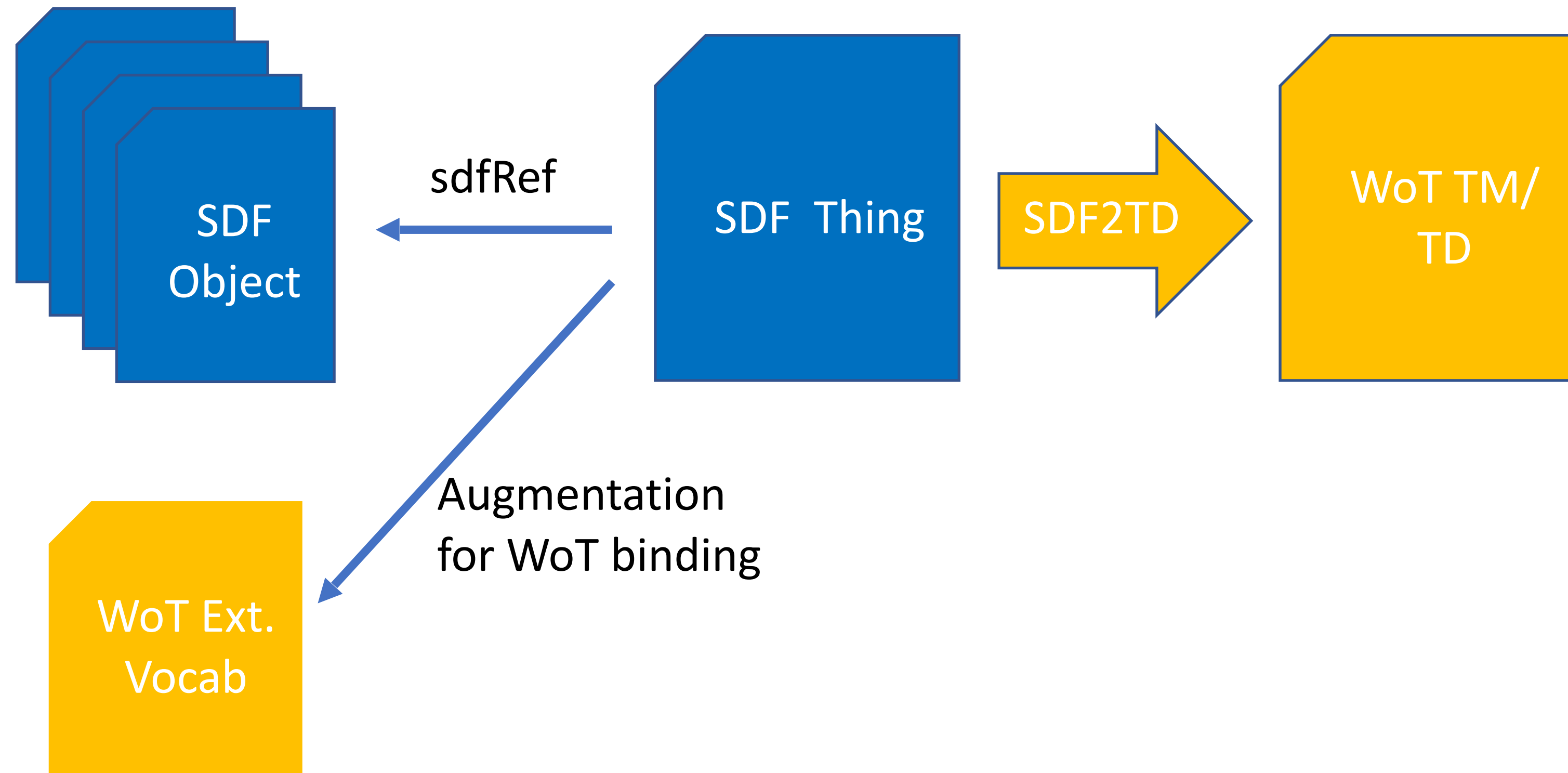
Design Alternatives

- The mapping file format could be an array of objects with the selector as an element in each object
- Mapping files could use JSON merge-patch type format and point to the root of a definition (not much different from JSON Pointer but could get hard to read)
- Selectors could be JSON Path
- Richer operations could be provided beyond augmentation and override

Thing model as a mapping file

- sdfThing class can be used to create instance-specific configurations of multiple objects and use mapping file design patterns in the same file
- Aggregation and customization in one file using SDF idioms
- Good readability and good authoring format

Example of Thing + Mapping



Next steps

- How should extension schemas work?
- Broad review and pressure test
- Formalize the design patterns
- sdf.next could introduce extension point schemas and mapping formats
- When should there be separate I-D documents?

Value Scaling

- Numeric constraints should mostly be applied in the instance binding
 - Unit may be constrained in the abstract model
 - Minimum, maximum, multiple, fixed/float type, and optional scaling should be left to the instance
 - Not usually expected to be applied in layers
 - Some playground models may be over-constrained
 - Some clear exceptions, e.g. % data may define minimum=0 and maximum=100 in the model but leave resolution and scaling to the instance binding

SDF compact form

Carsten Bormann

ASDF@IETF 110, 2021-03-11

Observations

- SDF has been designed and optimized for interchange between **tools**
- But **humans** have to work on SDF models, too
- Role model: Relax-NG started as an XML-based format
- Relax-NG compact (RNC) was designed for the whiteboard: human interaction with less noise and more focus on substance
- What would an SDF compact look like?

Input: A random SDF model from the playground

```
{
  "info": {
    "copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved.",
    "license": "https://github.com/one-data-model/oneDM/blob/master/LICENSE",
    "title": "Cadence",
    "version": "2019-06-11"
  },
  "sdfObject": {
    "cadence": {
      "description": "This Resource describes the cadence, which is the number of revolutions of crank per minute when cyclists pedal the pedals. The unit, which is the default unit, is rpm. The cadence Property is a read-only value that is provided by the server. When range (from \"oic.r.baseresource\") is omitted the default is 0 to +MAXFLOAT.",
      "sdfProperty": {
        "cadence": {
          "description": "This Property describes the rate at which a cyclist is pedalling/turning the pedals.",
          "type": "integer",
          "minimum": 0,
          "unit": "1/min",
          "writable": false
        },
        "range": {
          "type": "array",
          "description": "The valid range for the Property in the Resource as an integer. The first value in the array is the minimum value, the second value in the array is the maximum value.",
          "writable": false,
          "minItems": 2,
          "maxItems": 2,
          "items": {
            "type": "integer"
          }
        },
        "step": {
          "description": "Step value across the defined range when the range is an integer. This is the increment for valid values across the range; so if range is 0..10 and step is 2 then valid values are 0,2,4,6,8,10.",
          "writable": false,
          "type": "integer"
        }
      },
      "sdfRequired": [
        "#/sdfObject/cadence/sdfProperty/cadence"
      ]
    }
  }
}
```

Compact form

info:

```
copyright: Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved.  
version: '2019-06-11'  
title: Cadence  
license: https://github.com/one-data-model/oneDM/blob/master/LICENSE
```

object cadence:

```
:: This Resource describes the cadence, which is the number of  
   revolutions of crank per minute when cyclists pedal the  
   pedals. The unit, which is the default unit, is rpm. The cadence  
   Property is a read-only value that is provided by the  
   server. When range (from "oic. r. baseresource") is omitted the  
   default is 0 to +MAXFLOAT.
```

property cadence:

```
r-o!: integer .ge 0  
unit: 1/min
```

```
:: This Property describes the rate at which a cyclist is pedalling/turning  
   the pedals.
```

property range:

```
r-o?: [integer, integer]
```

```
:: The valid range for the Property in the Resource as an integer.  
   The first value in the array is the minimum value, the second value in the  
   array is the maximum value.
```

property step:

```
r-o?: integer
```

```
:: Step value across the defined range when the range is an integer. This  
   is the increment for valid values across the range; so if range is 0..10  
   and step is 2 then valid values are 0,2,4,6,8,10.
```

Design

- Let YAML format do most of the work
 - YAML is widely used as humane form of JSON
- Combine section name and item name into one label: object cadence
- Replace “description:” by “::”
- Merge readable/writable/observable, optionality, and data shape into one line:
`r-o?: [integer, integer]`

Compact form

info:

copyright: Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved.

version: '2019-06-11'

title: Cadence

license: <https://github.com/one-data-model/oneDM/blob/master/LICENSE>

object cadence

description → ::
This property is the number of revolutions per minute of the pedals. The unit, which is the default unit, is rpm. The cadence property is a read-only value that is provided by the server. When range is omitted the default is 0 to +MAX.

property cadence:

Label like a declaration

r-o!: integer range 0
unit: 1/min

:: This Property describes the rate at which a cyclist is pedalling/turning the pedals.

property range:

rwo? bits, data shape

r-o?: [integer, integer]


:: The valid range for the Property is the range. The first value in the array is the minimum value, the second value in the array is the maximum value.

property step:

r-o?: integer

:: Step value across the defined range when the range is an integer. This is the increment for valid values across the range; so if range is 0..10 and step is 2 then valid values are 0,2,4,6,8,10.

Next steps

- Complete the implementation
 - Verify it against playground models and others
 - Publish sdfc tool, play with it
 - Target informational status for draft-bormann-asdf-sdf-compact; publish with SDF spec
- 
- April timeframe