SDF: The Semantic Definition Format
A brief tutorial and status

ASDF WG @ IETF 118, 2023-11-06

Carsten Bormann

The need for One Data Model

- IoT standardization is dominated by **ecosystem**-specific SDOs
- Each ecosystem has their own data models,
  and their own way to document them
- IoT applications may need to work with *things* from multiple ecosystems:
  No single ecosystem can supply the whole variety needed
- Can build protocol translators; harder to translate **hundreds** of data models

SDF: The Semantic Definition Format

- https://github.com/ietf-wg-asdf
- Defines classes of *things* (sdfObject, combine into sdfThing)
- Things have **interactions** with their *clients(*)*,
  provided by **affordances**
- Interaction affordances grouped into **interaction patterns**:
  For now, **Property, Action, Event**
- Interactions input and output **data** (groupable into sdfData)

(*) Not an SDF term

Overall Structure of an SDF model

- One or more JSON documents; linked together with JSON pointers [RFC6901]
- An SDF specification can **reuse** elements (such as sdfData definitions) of other SDF specifications
- SDF “document” structured into groups/blocks

Interaction Patterns

- SDF is about modeling data
- Interaction Patterns mostly defined along input and output data

Name	cf. REST	Initiative	Input	Output
Property	GET	Client	—	Data
Property (writable)	PUT	Client	Data	(Data)
Action	POST	Client	Input	Output
Event	?	Thing	—	Output

Action

- Actions can have different input and output data
- Some actions take time (not modeled): Initiative to return output moved to Thing (~ Event)

Name	cf. REST	Initiative	Input	Output
Property	GET	Client	—	Data
Property (writable)	PUT	Client	Data	Data
Action	POST	Client	Input	Output
Event	?	Thing	—	Output

Property

- Property is used for data items that can be read by the client
- Writable properties can also be “set” (no special output)
- Observable properties look like an Event

Name	cf. REST	Initiative	Input	Output
Property	GET	Client	—	Data
Property (writable)	PUT	Client	Data	(Data)
Property (observable)	GET (observe)	Client, Thing	—	Data
Event	?	Thing	—	Output

Event

- Least well-defined interaction pattern
- Is an Event just a notification (similar to observable property)?
- Are Events just status updates (temperature) or is any single one of them precious (coin insertion)?

Name	cf. REST	Initiative	Input	Output
Property	GET	Client	—	Data
Property (writable)	PUT	Client	Data	Data
Action	POST	Client	Input	Output
Event	?	Thing	—	Output

Data

- Data is defined by their *shape* (as in data definition/“schema” languages)
- Data definitions can be made inline in an affordance definition or separately, for later reference
- Definitions can use curated **subset** of json-schema.org terms, and/or SDF-specific terms such as contentFormat, nullable, scale…
- Beyond SDF-base:
  Mapping information (**protocol bindings**) helps bind these data to ecosystem specific formats and encodings

Data Model vs. Information Model (1)

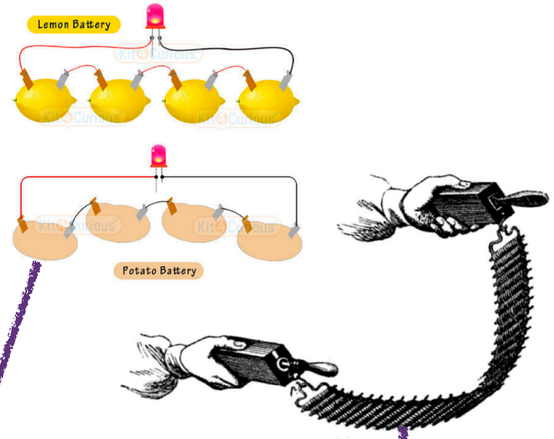
- SDF uses json-schema.org-style data modeling,
  enhanced by SDF qualities

- Really: This should be **information models** (RFC 3444):
  - Abstract from arbitrary representation decisions
  - Don’t commit to specific numbers, strings, etc. (**bindings** can do that)
  - Beyond SDF-base: Bind to **semantics** via RDF-style links

Data Model vs. Information Model (2)

- “Enums”: choices of values (strings, integers), each usually denoting some specific concept
- Information model: Don’t commit to specific representation (**bindings** can do that)
- Could bind to **semantics** via RDF-style links

- sdfChoice as a generalized choice construct

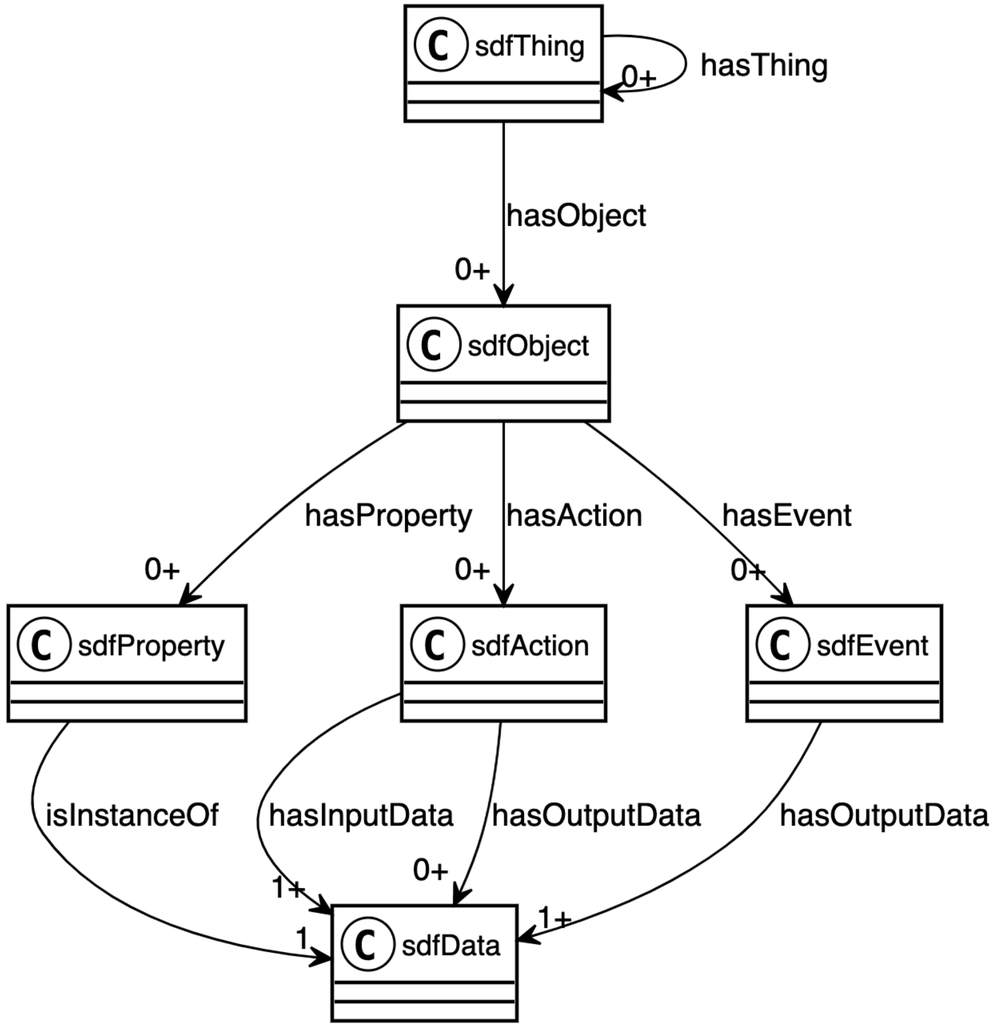


"Alkaline", "Aluminium Air", "Aluminium Ion", "Atomic Betavoltaics", "Atomic Optoelectric Nuclear", "Atomic Nuclear", "Bunsen Cell", "Chromic Acid Cell", "Poggendorff Cell", "Clark Cell", "Daniell Cell", "Dry Cell", "Earth", "Flow", "Flow Vanadium Redox", "Flow Zinc Bromine", "Flow Zinc Cerium", "Frog", "Fuel", "Galvanic Cell", "Gauss", "Grove Cell", "Lead Acid", "Lead Acid Deep Cycle", "Lead Acid VRLA", "Lead Acid AGM", "Lead Acid Gel", "Leclanche Cell", “**Lemon Potato**", "Lithium", "Lithium Air", "Lithium Ion", "Lithium Ion Cobalt Oxide (ICR)", "Lithium Ion Manganese Oxide (IMR)", "Lithium Ion Polymer", "Lithium Iron Phosphate", "Lithium Sulfur", "Lithium Titanate", "Lithium Ion Thin Film", "Magnesium", "Magnesium Ion", "Mercury", "Molten Salt", "Nickel Cadmium", "Nickel Cadmium Vented Cell", "Nickel Hydrogen", "Nickel Iron ", "Nickel Metal Hydride", "Nickel Metal Hydride Low Self-Discharge", "Nickel Oxyhydroxide", "Nickel Oxyride", "Nickel Zinc", "Organic Radical", "Paper", "Polymer Based", "Polysulfide Bromide", "Potassium Ion", "**Pulvermachers Chain**", "Silicon Air", "Silver Calcium", "Silver Oxide", "Silver Zinc", "Sodium Ion", "Sodium Sulfur", "Solid State", "Sugar", "Super Iron", "UltraBattery", "Voltaic Pile", "Voltaic Pile Penny", "Voltaic Pile Trough", "Water Activated", "Weston Cell", "Zinc Air", "Zinc Carbon", "Zinc Chloride", "Zinc Ion", "Unknown"

OCF “oic.r.batterymaterial”

sdfThing

- sdfObject definitions can be combined into top-level structures
- sdfThing can contain sdfObject and sdfThing



[draft-onedm-t2trg-sdf-00]

Specifying SDF

- SDF documents are JSON texts
  Structure (grammar) can be specified in a data definition language
  - Using CDDL:
    https://github.com/one-data-model/SDF/blob/master/sdf-feature.cddl
  - Translated into json-schema.org “JSON Schema” format:
    https://github.com/one-data-model/SDF/blob/master/sdf-framework.jso.json and
    https://github.com/one-data-model/SDF/blob/master/sdf-validation.jso.json
    (do not confuse with the selected json-schema.org terms used **inside** SDF)
- Of course, also needs semantics (in English text);
  - Definition: https://github.com/ietf-wg-asdf
- De-facto specifics via tooling (e.g., at https://github.com/one-data-model/tools)

Status 2023-11-06

- Original OneDM SDF specification 2020-06-05 (draft-onedm-t2trg-sdf-00)
  - Now ~ 200 data models in playground, exploratory, unit_test repos
  - Ecosystem SDOs have developed tools to convert their corpus to SDF
- ASDF WG picked up OneDM spec, developed into **SDF-base**
  - WGLC completed 2023-09-20,
    grace period added, –17 submitted yesterday