

WISHI

March 2nd (Tuesday), 15:00–16:00 UTC
(16:00–17:00 CEST, 07:00–08:00 PDT)

Draft agenda

Start	Topic	Who	ΔT
1500Z	IETF ASDF/WISHI Hackathon (, 20 min)	Carsten	20
1520Z	Semantic technology landscape update	Milan M & Michael K	10
1530Z	WISHI way forward: — Future topics related to SDF — Other semantics & hypermedia topics	Ari & all	20
1550Z	Future WISHI meetings	Ari	5

ASDF/WISHI Hackathon

- validate SDF1.1, move tools and models up to SDF1.1
- continue work on the "semantic proxy"
(connect OMA and OCF clients/servers guided by common SDF)
- continue development of the "mapping file" concept (below)
- "convert" between WoT TD and SDF,
further developing instance-specific mapping files
- derive clarifications for the processing model:
 - how do "cross-links" work, external references using the namespace

Coordination calls

- Mon 1500Z (before OneDM)
- Tue 1600Z (following WISHI)
- Wed 1630Z (following IoTDIR)
- Thu 1500Z or 1600Z

via WebEx — also permanent meeting room

Wiki, day-to-day Notes

Mapping files

- ecosystem specific mapping files
 - (exploratory repo): example that defines representation (data model level) data types, as well as assigns IDs (see here for worked example for the latter)
- instance specific mapping files
 - protocol parameters (IP address etc.)
 - non-technical instance information (physical context, purpose)
 - data models for instance mappings (!)
- how are mapping files “activated”?
How do they come into force for a specific environment/instance?

Data Schemas for Mapping Files

SDF is confusing, as we have

- 2 Model in JSON, with a formal description (CDDL/JSO) in document (meta-model)
- 1 Data schemas in the model, for properties and action/event I/O

Now getting more confusing, as we have

- 2 Model in JSON, structure specific to the mapping desired (needs meta-model)
- 1 model contains data schemas like the main SDF models
- 3 needs meta-meta-model for meta-models in 2, specified in document!

Modeling vs. Transformation

Ecosystem-specific model information in mapping files

- Define parameters that need to be added to abstract SDF model for protocol binding
 - instance information; model of that instance information
 - invariant information (e.g., IPSO IDs)
- Since mapping models map, need description of the mapping
 - map abstract SDF model (information model) data to ecosystem-specific data model
 - easy: linear transformation (°C to °F)
 - not so easy: non-linear relationships, structural transforms, ...

Example: Ecosystem-specific static mapping info

onedm/exploratory/strawman-examples/ZCL/zcl.smf.json

```
"sdfMap": {  
  "ex:#/sdfObject/Level" : {  
    "id": 8  
  },  
}
```

whose id is that? Where is the metamodel that says what shape the IDs have?

could get away with always having this mapping active

Example: Ecosystem-specific static transformation info

onedm/exploratory/strawman-examples/ZCL/zcl.smf.json

```
"ex:#/sdfObject/Level/sdfData/MoveStepMode/sdfEnum/Up" : {  
  "type": "number",  
  "const": 0  
},  
"ex:#/sdfObject/Level/sdfData/MoveStepMode/sdfEnum/Down" : {  
  "type": "number",  
  "const": 1  
}
```

works great with sdfChoice (each choice has a JSON pointer)
how to map from °C to °F (and generalize that)?

need to specifically activate this mapping!

Example: Ecosystem-specific static transformation info

t2trg/2021-03-hackathon/mjkoster/sdfthing-modbus-dcpowersupply.sdf.json

```
"VoltageMeasurement": {
  "sdfRef": "pg:#/sdfObject/Voltage",
  "sdfProperty": {
    "Sensor_Value": { [...]
      "sdfRef": "#/sdfThing/DCPowerSupply_LW3010E/sdfData/VoltageData",
      "DataSchema": {
        "WidthInBits": 16
      },
      [...]
    }
  }
},
```

How is the voltage range mapped into the 2^{16} values?
(Parameters for linear transformation)

Example: Ecosystem-specific protocol binding info

t2trg/2021-03-hackathon/mjkoster/sdfthing-modbus-dcpowersupply.sdf.json

```
"VoltageMeasurement": {
  "sdfRef": "pg:#/sdfObject/Voltage",
  "sdfProperty": {
    "Sensor_Value": {
      [...]
      "WotProtocolBinding": {
        "href": { "sdfRef": "#/sdfThing/LW3010E/sdfData/ThisIPAddress/" },
        "modv:unitID": { "sdfRef": "#/sdfThing/LW3010E/sdfData/ThisUnitID" },
        "modv:entity": "HoldingRegister",
        "modv:offset": "1002",
        "modv:length": 1
      }
    }
  }
},
```

Example: Ecosys-specific protbind info: reversed?

[gist.../JKRhb/e3def2cf049e90e43f533132b6fb7467/sdf-binding-reversed.json](https://gist.github.com/JKRhb/e3def2cf049e90e43f533132b6fb7467/sdf-binding-reversed.json)

```
"sdfBindings": {
  "coap": {
    "defaultSecurity": "psk_sc",
    "port": "5683",
    "/switch/value": {
      "readSdfProperty": {
        "method": "GET",
        "affordance": "sdfObject/Switch/sdfProperty/value",
        "security": "nosec_sc"
      },
      "writeSdfProperty": {
        "method": "PUT",
        "affordance": "sdfObject/Switch/sdfProperty/value"
      },
      [...]
    },
    "/switch/on": {
      "invokeSdfAction": {
        "method": "POST",
        "affordance": "sdfObject/Switch/sdfAction/on"
      }
    }
  },
}
```

Todo

Need to understand processing model

Need to make strawman examples like the above

Use these to understand the elements of the meta-model