Playing with DTDL

* https://github.com/Azure/opendigitaltwins-dtdl/tree/master/DTDL/v2/samples

```
{
  "@context": "dtmi:dtdl:context;2",
  "@id": "dtmi:com:example:Thermostat;1",
  "@type": "Interface",
  "displayName": "Thermostat",
  "description": "Reports current temperature and provides desired temperature control.",
```

```
{
  "contents": [
    {
      "@type": [
        "Telemetry",
        "Temperature"
      ],
      "name": "temperature",
      "displayName" : "Temperature",
      "description" : "Temperature in degrees Celsius.",
      "schema": "double",
      "unit": "degreeCelsius"
    },
    {
      "@type": [
        "Property",
        "Temperature"
      ],
      "name": "targetTemperature",
      "schema": "double",
      "displayName": "Target Temperature",
      "description": "Allows to remotely specify the desired target temperature.",
      "unit" : "degreeCelsius",
      "writable": true
    },
```

```
{
  "@type": [
    "Property",
    "Temperature"
  ],
  "name": "maxTempSinceLastReboot",
  "schema": "double",
  "unit" : "degreeCelsius",
  "displayName": "Max temperature since last reboot.",
  "description": "Returns the max temperature since last device reboot."
}
```

```
{
  "@type": "Command",
  "name": "getMaxMinReport",
  "displayName": "Get Max-Min report.",
  "description": "This command returns the max, min and average temperature...",
  "request": {
    "name": "since",
    "displayName": "Since",
    "description": "Period to return the max-min report.",
    "schema": "dateTime"
  }
```

```
{
      "response": {
        "name" : "tempReport",
        "displayName": "Temperature Report",
        "schema": {
          "@type": "Object",
          "fields": [
            { "name": "maxTemp", "displayName": "Max temperature", "schema": "double"
            },
            { "name": "minTemp", "displayName": "Min temperature", "schema": "double"
            },
            { "name" : "avgTemp", "displayName": "Average Temperature", "schema": "double"
            },
            { "name" : "startTime", "displayName": "Start Time", "schema": "dateTime"
            },
            { "name" : "endTime", "displayName": "End Time", "schema": "dateTime"
            }
          ]
        }
      }
    }
```

Terminology

SDF	DTDL
sdfProperty	Property
sdfAction	Command
sdfEvent	Telemetry
sdfObject	Component
sdfInputData	request
sdfOutputData	response
label	displayName
description	description
units	unit
sdfInputData	request (in Command)
sdfOutputData	response (in Command)
type	schema

Types and units

SDF types from JSON schema. SDF units from SenML units registry.

SDF	DTDL
number	double
integer	integer
date-time (format)	dateTime (type)

Cel	degreeCelsius

SDF

```
{
  "info" : {
    "title" : "Azure DTDL TemperatureController in SDF",
    "version" : "2020-07-30", "license": "TBD", "copyright": "TBD"
  },
  "sdfThing": {
    "TemperatureController": {
      "label": "Temperature Controller",
      "description": "Device with two thermostats and remote reboot.",
      "sdfObject" : {
        "TemperatureController" : {
          "$comment" : "Wrapping controller level E/A/P here",
          "sdfEvent" : {
            "workingSet" : {
              "label" : "Working Set",
              "description": "Current working set of the device memory in KiB.",
              "sdfOutputData" : ["#/sdfData/workingSet"]
            }
          },
          "sdfProperty" : {
            "serialNumber" : {
              "label": "Serial Number",
              "description": "Serial number of the device.",
              "type": "string"
            }
          },
```

```
{ "...": {
          "sdfAction" : {
            "reboot" : {
              "label": "Reboot",
              "description": "Reboots the device after waiting the number of seconds specified.",
              "sdfInputData" : ["#/sdfData/since"]
            }
          }
        },
        "Thermostat1" : {
          "sdfRef" : "#/sdfObject/Thermostat",
          "label" : "Thermostat One",
          "description": "Thermostat One of Two."
        },
        "Thermostat2" : {
          "sdfRef" : "#/sdfObject/Thermostat",
          "label" : "Thermostat Two",
          "description": "Thermostat Two of Two."
        },
        "deviceInformation" : {
          "sdfRef" : "TBD: dtmi:azure:DeviceManagement:DeviceInformation;1",
          "label" : "Device Information interface",
          "description": "Optional interface with basic device hardware information."

        }
      }
```

```
{
  "sdfObject": {
    "Thermostat": {
      "sdfEvent": {
        "Temperature": {
          "description": "Temperature in degrees Celsius.",
          "sdfOutputData" : ["#/sdfData/temperature"]
        }
      },
      "sdfProperty": {
        "targetTemperature": {
          "label": "Target Temperature", "description": "Allows to remotely specify the desired target temperature.",
          "type": "number", "units": "Cel",
          "writable": true
        },
        "maxTempSinceLastReboot": {
          "label": "Max temperature since last reboot.", "description": "Returns the max temperature since last device reboot.",
          "type": "number", "units": "Cel"
        }
      },
      "sdfAction": {
        "getMaxMinReport": {
          "label": "Get Max-Min report.", "description": "This command returns the max, min and average temperature... time.",
          "sdfInputData": ["#/sdfData/since"],
          "sdfOutputData": [
            "#/sdfData/maxTemp", "#/sdfData/minTemp", "#/sdfData/avgTemp", "#/sdfData/startTime", "#/sdfData/endTime"
          ]
        }
      }
    }
  },
```

```
{
  "sdfData": {
    "workingSet" : {
      "type": "number", "units" : "KiB"
    },
    "delay" : { "label" : "Delay", "description": "Number of seconds to wait before rebooting the device.",
      "type" : "integer"
    },
    "since" : { "label": "Since", "description": "Period to return the max-min report.",
      "type" : "string", "format": "date-time"
    },
    "temperature" : {
      "type": "number", "units": "Cel"
    },
    "maxTemp": { "label": "Max temperature",
      "type": "number"
    },
    "minTemp": { "label": "Min temperature",
      "type": "number"
    },
    "avgTemp": { "label": "Average Temperature",
      "type": "number"
    },
    "startTime" : { "label": "Start Time",
      "type" : "string", "format": "date-time"
    },
    "endTime": { "label": "End Time",
      "type" : "string", "format": "date-time"
    }
  }
}
```

DTDL/WISHI: Next Steps

Do a bit of tools hacking
to make this one-off example more repeatable

A canon of DTDL models would help.

Learn from this, and discuss at the next WISHI meeting?

Next Meetings

Keep up ~ monthly cadence

Pause in August

Plan next meeting for mid-september, W38

Doodle to be sent to t2trg@irtf.org