WoT Virtual F2F Thing Description

Sebastian Kaebisch

24 March 2021

Agenda

- Whats new in TD 1.1? (20m)

- Aim for next TD publications
  - Discuss (10min)

- Thing Model (1h30m, 10min break in between)
  - required feature
  - import feature
  - file extensions and media type
  - feedback from PlugFest about SDF-TM usage (Michael Koster)

- TD Canonicalization (20m) (Michael Lagally / Michael McCool)

- News from iotschema.org (Michael Koster/Darko Anicic)

What's new in TD 1.1?
Diff from REC 1.0 to FPWD 1.1 (vF2F Oct 2020)

Please check

https://github.com/w3c/wot/blob/main/PRESENTATIONS/2020-10-online-f2f/2020-10-21-TD-Topics-Kaebisch.pdf

What's new in TD 1.1?
observeallproperties & unobserveallproperties

op	Indicates the semantic intention of performing the operation(s) described by the form. For example, the Property interaction allows get and set operations. The protocol binding may contain a form for the get operation and a different form for the set operation. The op attribute indicates which form is for which and allows the client to select the correct form for the operation required. op can be assigned one or more interaction verb(s) each representing a semantic intention of an operation.	optional	any type (one of readproperty, writeproperty, observeproperty, unobserveproperty, invokeaction, subscribeevent, unsubscribeevent, readallproperties, writeallproperties, readmultipleproperties, writemultipleproperties, observeallproperties, or unobserveallproperties)

https://w3c.github.io/wot-thing-description/#form

What's new in TD 1.1?
init rel types in links

Value	Occurrence	Explanation	Source of value origin
service-doc	0..*	Relation to a resource that provide (human-readable) documentation or descriptions.	IANA Link Relation
alternate	0..*	Point to alternative representation of the Thing (i.e. RDF-Turtle, human-readable HTML document, ...).	IANA Link Relation
type	0..1	Indicate that the Thing is an instance of the target resource such as to a Thing Model.	IANA Link Relation
extends	0..1	Extends an existing definition of the targed resource such as a Thing Model.	W3C WoT Thing Model
manifest	0..*	Point to a software implementation of the TD.	IANA Link Relation
proxy-to	0..*	Target resource provide the address of a proxy.	W3C WoT Security and WoT Binding Template
collection	0..1	Points to a collections of Things.	IANA Link Relation
item	0..*	Points to a Thing that is member of the current Thing collections.	IANA Link Relation
predecessor-version	0..1	Points to a previous Thing Description or Thing Model version.	IANA Link Relation
controlledBy	0..*	Refers to a Thing that controls the context Thing.	W3C Thing Description

https://w3c.github.io/wot-thing-description/#link

W3C Web of Things (WoT) WG/IG

What's new in TD 1.1?
exclusiveMinimum & exclusiveMaximum

exclusiveMinimum	Specifies a minimum numeric value, representing an exclusive lower limit. Only applicable for associated number or integer types.	optional
maximum	Specifies a maximum numeric value, representing an inclusive upper limit. Only applicable for associated number or integer types.	optional
exclusiveMaximum	Specifies a maximum numeric value, representing an exclusive upper limit. Only applicable for associated number or integer types.	optional

IntegerSchema

minimum : integer	
exclusiveMinimum	: integer
maximum: integer	
exclusiveMaximum	: integer
multipleOf	: integer

NullSchema

NumberSchema

minimum : double	
exclusiveMinimum	: double
maximum: double	
exclusiveMaximum	: double
multipleOf	: double

https://w3c.github.io/wot-thing-description/#numberschema

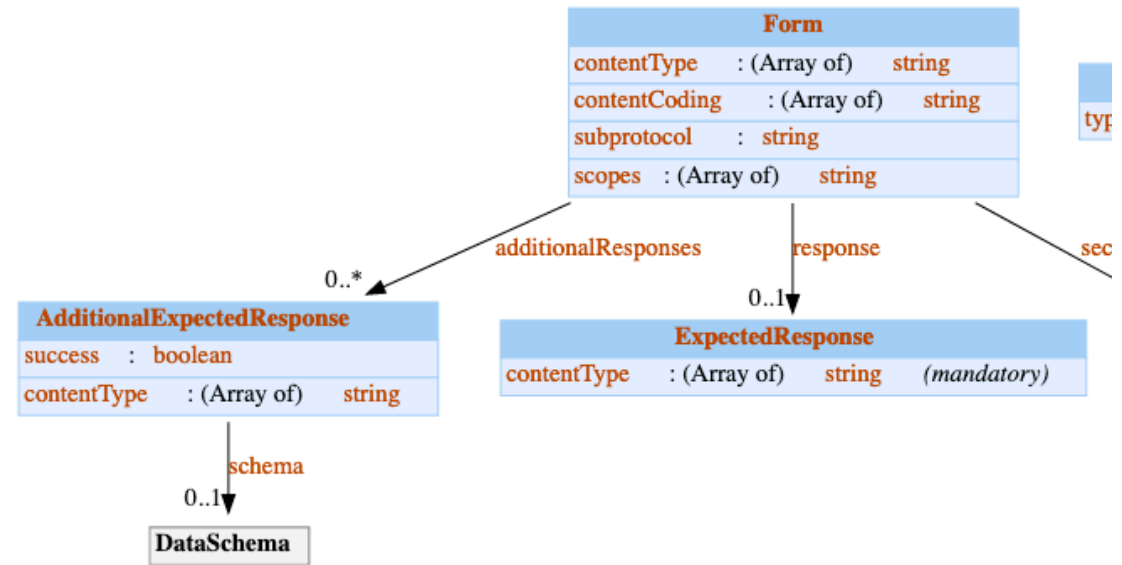https://w3c.github.io/wot-thing-description/#integerschema

What's new in TD 1.1?
AdditionalExpectedResponse

§ 5.3.4.4 *AdditionalExpectedResponse*

Communication metadata describing the expected response message for additional responses.

Vocabulary term	Description	Assignment	Type
success	Signals if an additional response should not be considered an error.	with default	boolean
schema	Used to define the output data schema of the Interaction for an additional response.	optional	DataSchema
contentType	Assign a content type based on a media type (e.g., text/plain) and potential parameters (e.g., charset=utf-8) for the media type [RFC2046].	optional	string or Array of string

Form

contentType	: (Array of)	string	
contentCoding	: (Array of)	string	
subprotocol	: string		
scopes	: (Array of)	string	

additionalResponses 0..*

response 0..1

sec

typ

AdditionalExpectedResponse

success	: boolean		
contentType	: (Array of)	string	

schema 0..1

DataSchema

ExpectedResponse

contentType	: (Array of)	string	*(mandatory)*

https://w3c.github.io/wot-thing-description/#additionalexpectedresponse

What's new in TD 1.1?
uri assignment for authentication location

§ 5.3.3.4 *BasicSecurityScheme*

Basic Authentication [RFC7617] security configuration identified by the Vocabulary Term basic (i.e., "scheme": "basic"), using an unencrypted username and password. This scheme should be used with some other security mechanism providing confidentiality, for example, TLS.

Vocabulary term	Description	Assignment	Type
name	Name for query, header, cookie, or uri parameters.	optional	string
in	Specifies the location of security authentication information.	optional	string (one of header, query, body, cookie, or uri)

§ 5.3.3.5 *DigestSecurityScheme*

Digest Access Authentication [RFC7616] security configuration identified by the Vocabulary Term digest (i.e., "scheme": "digest"). This scheme is similar to basic authentication but with added features to avoid man-in-the-middle attacks.

Vocabulary term	Description	Assignment	Type
name	Name for query, header, cookie, or uri parameters.	optional	string
qop	Quality of protection.	optional	string (one of

…

For all security schemes, any private keys, passwords, or other sensitive information directly providing access should be shared and stored out-of-band and *MUST NOT* be stored in the TD. The purpose of a TD is to describe how to access a Thing if and only if a Consumer already has authorization, and is not meant be used to grant that authorization.

Security schemes generally may require additional authentication parameters, such as a password or key. The location of this information is indicated by the value associated with the name in, often in combination with the value associated with name. The value associated with in can take one of the following values:

header:
The parameter will be given in a header provided by the protocol, with the name of the header provided by the value of name.

query:
The parameter will be appended to the URI as a query parameter, with the name of the query parameter provided by name.

body:
The parameter will be provided in the body of the request payload, with the data schema element used provided by name. If name is not given, it is assumed the entire body is to be used as the security parameter.

cookie:
The parameter is stored in a cookie identified by the value of name.

uri:
The parameter is embedded in the URI itself, which is encoded in the relevant interaction using a URI template variable defined by the value of name. This is more general than the query mechanism but more complex. The value uri *SHOULD* be specified for the name in in a security scheme only if query is not applicable. The URIs provided in interactions where a security scheme using uri as the value for in *MUST* be a URI template including the defined variable.

https://w3c.github.io/wot-thing-description/#securityscheme

What's new in TD 1.1?
TM Features

```json
{
    ...
    "@type" : "ThingModel",
    "title": "Lamp Thing Model",
    "description": "Lamp Thing Description Model",
    "version" : {"model" : "1.0.0" },
    ...
}
```

```json
"title": "Smart Lamp Control",
"links" : [{
    "rel": "extends",
    "href": "http://example.com/BasicOnOffTM",
    "type": "application/td+json"
}],
"properties" : {
```

```json
"title": "Thermostate No. {{THERMOSTATE_NUMBER}}",
"base": "mqtt://{{MQTT_BROKER_ADDRESS}}",
"properties": {
```

```json
"properties": {
    "status": {
        "description": "current status of the lamp (on|off)",
        "type": "string",
        "readOnly": true
    },
    "required": ["status"]
},
```

What's new in TD 1.1?
TM Features

- All definitions in the applied Thing Model have to taken over in the Thing Description instance. If used, the extension and imports feature according to Section § 10.3.2 Extension and Import must be resolved and represented in the Thing Description instance as well.

- The value `ThingModel` value of the top-level `@type` have to be replaced by the value `Thing` in the Thing Description instance.

- If the `required` feature is used based on Section § 10.3.4 Required, the required interactions must definitly be taken over in the Thing Description instance.

- If used, all placholder (see Section § 10.3.3 Placeholder) in a Thing Model must be replaced with a valid corresponding value in the Thing Description.

- Missing communication and/or security metadata details have to be complemented in the Thing Description instance based on Section § 6.3.4 `securityDefinitions` and `security` and/or § 6.3.9 `forms`.

- If used, `links` entry with `"rel":"extends"` should not taken over in the Thing Description instance.

https://w3c.github.io/wot-thing-description/#thing-model
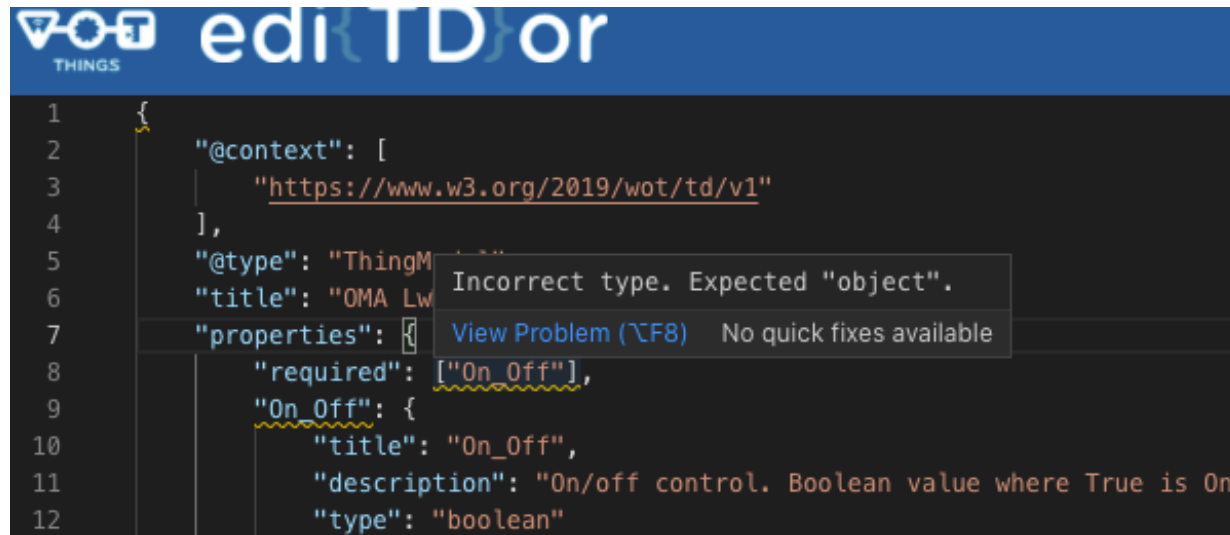
Roadmap for next Publication

- Next WD (until End of April / middle May) should cover
  - missing TM feature → @sebastiankb
  - multipleResponses alternative → @mmcool
  - Labeled "PR needed" (https://github.com/w3c/wot-thing-description/issues?q=is%3Aissue+is%3Aopen+label%3A%22PR+needed%22)
  - Profile identification
  - Validation & Canonization?

- CR (June/July; depending on WoT Profile)
  - fix all semantic based issues (also discovery dependent) → @victor
  - Maybe some related issues regarding profile (e.g., operations in a top level form, History etc.)
  - Provide assertion risks (depends of the PlugFest)

TM Issues - required at interaction level Problem Statement

Confusing for TD/TM processor that the **required** term is showing up at interaction level. Validator showing an error since it is not a normal interaction definition.

TM Issues - required at interaction level
Solution Proposal

- Current approach is adopted from JSON Schema concept

- SDF defines **sdfRequired**
  https://ietf-wg-asdf.github.io/SDF/sdf.html#section-4.5
  → "The value of "sdfRequired" is an array of name references, each pointing to one declaration the instantiation of which is declared mandatory."
  → is defined outside of the interaction

```json
{
  "sdfObject": {
    "temperatureWithAlarm": {
      "sdfRequired": [
        "#/sdfObject/temperatureWithAlarm/sdfProperty/currentTemperature",
        "#/sdfObject/temperatureWithAlarm/sdfEvent/overTemperatureEvent"
      ],
      "sdfData":{
        "temperatureData": {
          "type": "number"
        }
      },
      "sdfProperty": {
        "currentTemperature": {
          "sdfRef": "#/sdfObject/temperatureWithAlarm/sdfData/temperatureData"
        }
      },
      "sdfEvent": {
        "overTemperatureEvent": {
          "sdfOutputData": {
```

TM Issues - required at interaction level Solution Proposal

- Follow same approach as **sdfRequired**

- Name proposals: tdRequired / tmRequired / wotRequired...

- define outside of the interactions:

```
...
"security": ["nosec_sc"],
  "tmRequired" : ["#/properties/homeLoc"],
  "properties": {
    "homeLoc": {
      "title": "Home location",
      "description": "Home location, can be set to a new value. goHome action goes there",
      "type": "object",
...
```

TM Issues – import mechanism
Problem Statement

We can extend a TM which inherit all definitions.

However, we cannot import sub-definition of one or more TMs.
E.g., we only interested in status property definition.

```json
[
    "@context": ["http://www.w3.org/ns/td"],
    "@type" : "ThingModel",
    "title": "Lamp Thing Model",
    "description": "Lamp Thing Description Model",
    "properties": {
        "status": {
            "description": "current status of the lamp (on|off)",
            "type": "string",
            "readOnly": true
        }
    },
    "actions": {
        "toggle": {
            "description": "Turn the lamp on or off"
        }
    },
```

```json
{
    "@context": ["http://www.w3.org/ns/td"],
    "@type" : "ThingModel",
    "title": "MyLamp",
    "links" : [{
        "rel": "extends",
        "href": "http://example.com/LampTM",
        "type": "application/td+json"
    }],
```

TM Issues - import mechanism
Solution Proposal

- JSON Schema concept has the **$ref** concept
  → mainly points to JSON Schema

- SDF defines **sdfRef**
  https://ietf-wg-asdf.github.io/SDF/sdf.html#section-4.4
  → "Copy all of the qualities of the referenced definition, indicated by the included name reference, into the newly formed definition.."

```
"temperatureProperty": {
  "sdfRef": "#/sdfData/temperatureData"
}
```

TM Issues - required at interaction level
Solution Proposal

- Follow same approach as **sdfRef**

- Name proposals: tdRef / tmRef / wotRef…

```json
{
    "@context": ["http://www.w3.org/ns/td"],
    "@type" : "ThingModel",
    "title": "My Lamp Thing Model",
    "properties": {
        "status": {
            "tmRef": "http://example.com/LampTM.tm.jsonld#/properties/status"
        }
    }
}
```

File Extension / Media Type
Open Questions

- we have no media type definition for TM
  → question is this needed or shall we use application/td+json or application/jsonld

- we need also clarification about a common file extension
  (e.g., td.jsonld and tm.jsonld)