Overview

IOT SCHEMA
The Problem

• Many standards organizations for connected things:
  – OCF, Zigbee, Z-Wave, Bluetooth, Fairhair
  – Focus on Device Certification
  – Exclusive, require membership to participate
  – Lack focus on common interoperability
  – Compete with each other, focus on verticals

• Each defines a unique device level application layer with dedicated data models, but...
  – Similar high level design patterns
  – Converging on common communication protocols (IPV6, CoAP, et. al.)
What needs to be built?

• Application level semantic interoperability
  – Well known formats to describe common affordances of connected things (What does it do? What can I control?)
  – A way to describe how to interact with connected things from different device ecosystems, which use similar protocols but diverse data models
  – Enable easy implementation of Bridges, Libraries, Translators, Mappings, Bindings, Proxies
Capability Abstraction

• Common abstraction => "Capability"
• A capability is the set of affordances needed to interact with a single function of a connected thing
• For example, an on/off switch capability
  – on/off state, delay time: **properties**
  – "switch on", "switch off": **actions**
  – "switched on", "switched off": **events**
Remarks:
• Feature of Interest (FoI) pattern is still to be integrated in the model (consider the integration of FoI into schema.org):
Example: LightControl Capability

```json
{
  "@id": "iot:LightControl",
  "@type": "rdfs:Class",
  "rdfs:subClassOf": {
    "@id": "iot:Capability"
  },
  "rdfs:comment": "A capability for controlling a light source, such as an RGB or other light",
  "rdfs:label": "LightControl",
  "iot:domain": [ {
    "@id": "iot:Home"}, {"@id": "iot:Building"} ],
  "iot:providesInteractionPattern": [ {
    "@id": "iot:BinarySwitch",
    "@id": "iot:SwitchStatus",
    "@id": "iot:TurnOn",
    "@id": "iot:TurnOff",
    "@id": "iot:CurrentColour",
    "@id": "iot:SetColour",
    "@id": "iot:CurrentDimmer",
    "@id": "iot:SetDimmer",
    "@id": "iot:RampTime"
  } ]
}
```

All interactions are optional

Capabilities are extensible
Example: SetDimmer Interaction

```
{
    "@id": "iot:SetDimmer",
    "@type": "rdfs:Class",
    "iot:acceptsInputData": {
        "@id": "iot:DimmerData"
    },
    "rdfs:comment": "Set quantized representation for brightness (e.g., in the range 0-100)",
    "rdfs:label": "SetDimmer",
    "rdfs:subClassOf": {
        "@id": "iot:Action"
    }
}
{
    "@id": "iot:DimmerData",
    "@type": "rdfs:Class",
    "rdfs:comment": "Dimmer data",
    "rdfs:label": "DimmerData",
    "rdfs:subClassOf": {
        "@id": "schema:PropertyValue"
    },
    "schema:propertyType": {
        "@id": "schema:Integer"
    },
    "schema:minValue": "schema:Integer",
    "schema:maxValue": "schema:Integer"
}
```
```json
{
"name": "LightControl",
"@type": ["Thing", "iot:LightControl"],
"base": "http://example.com:8080/mylamp",
"domain": ["iot:Building"],
"interaction": [ 
  {"name": "SwitchStatus", 
   "@type": [ 
       "Property",
       "iot:SwitchStatus"
     ],
   "observable": false,
   "schema": "boolean",
   "writable": false,
   "form": [ 
       {"href": "/switch",
        "mediaType": "application/ld+json"}
     ]
  }
]
}
```
Example: Annotated Thing Description

```json
{"@context": ["https://w3c.github.io/wot/w3c-wot-td-context.jsonld", "iot": "http://iotschema.org/"],
"name": "LightControl",
"@type": ["Thing", "iot:LightControl"],
"base": "http://example.com:8080/ylamp",
"domain": ["iot:Building"],
"interaction": [
  {
    "name": "SwitchStatus",
    "@type": ["Property", "iot:SwitchStatus"],
    "observable": false,
    "schema": "boolean",
    "writable": false,
    "form": [
      {
        "href": "/switch",
        "mediaType": "application/json"
      }
    ]
  }
]}
```
What is new in iot.schema.org

• The schema contains more terms
• Proposal on how to use shape constraints
• Few Event specifications have been added
• Few Action specifications (for writable Properties) have been added
• Attributes “writable” and “observable” added for Properties
Use of iot.schema.org

DISCOVERY
Motivation: Thing Discovery

- Problem to solve
  - Discover Things suitable for a WoT application;
Motivation: Thing Discovery

• Problem to solve
  – Discover Things suitable for a WoT application;
Motivation: Thing Discovery

- Problem to solve
  - Discover Things suitable for a WoT application;
Capability
Canonical URL: http://iotschema.org/Capability

Capability
Capability Class

<table>
<thead>
<tr>
<th>Property for InteractionPattern</th>
<th>Interaction Pattern</th>
<th>Description</th>
</tr>
</thead>
</table>

More specific Types

- Temperature Capability
- **Air conditioner Capability**
- Humidity Capability
- Illuminance Capability
- Light Control
- Motion Control
- Thermostat Capability

A property that relates a capability with its interaction patterns.
Air Conditioner Uniquely Identifiable
Instances of Property may appear as values for the following properties

<table>
<thead>
<tr>
<th>Property</th>
<th>On Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observes</td>
<td>Sensor</td>
<td>Relation between a Sensor and either a Property or an Event that it is capable of sensing.</td>
</tr>
<tr>
<td>For Property</td>
<td>Actuator</td>
<td>Relation between an Actuator and either a Property or an Action that it is capable of actuation.</td>
</tr>
</tbody>
</table>

More specific Types

- Motion Detected
- Turn On
- **Temperature**
- Current Level
- Turn Off
- Humidity
- **Binary Switch**
- Light Colour
- Operation Status
- Motion Type
- Switch Status
- Ramp Time
- Illuminace
- Set Dimmer
- Target Humidity
- **Target Temperature**
- Transition Time
- Run Mode
- Set Dimmer
- Count Down
- Light Colour
- Wind Strength
- Motion Detected

Interaction Patterns of AirConditioner Capability
Use of iot.schema.org

TD TEMPLATES
Thing Description Template Generation

- Automated generation of semantically annotated TDs
- Semantic validation of TD variations.

Steps:
1. Select Capabilities, interactions, and data from iotschema.org
2. Specify shape constraints for your Thing
3. Generate a TD
Example: Level Capability

```json
{
    "@context": [...],
    "@id": "iot:LevelCapability",
    "rdfs:subClassOf": "iot:Capability",
    "iot:providesInteractionPattern": [
        {
            "@id": "iot:CurrentLevel",
            "iot:providesOutputData": {
                "@id": "iot:LevelData",
                "schema:propertyType": "schema:Number",
                "schema:minValue": "schema:Number",
                "schema:maxValue": "schema:Number",
                "schema:unitCode": "qudt:Centimetre"
            },
            "rdfs:subClassOf": "iot:Property"
        },
        {
            "@id": "iot:SetLevel",
            "iot:acceptsInputData": "iot:LevelData",
            "iot:providesOutputData": "iot:LevelData",
            "rdfs:subClassOf": "iot:Action"
        }
    ]
}
```
SHACL Shape for Level Capability

Why Shapes are required on IoT capabilities?

- Variations between Things with same Capability
- Validate semantically-enriched Thing Descriptions
- Shared documentation between manufacturers (enhances interoperability)

`:LevelCapabilityShape
  a sh:NodeShape ;
  sh:targetClass :LevelCapability ;
  sh:property [ sh:path :providesInteractionPattern ; sh:minCount 1 ; sh:maxCount 1 ; sh:in (:CurrentLevel) ; ] ; .

`:CurrentLevelShape
sh:targetClass :CurrentLevel ;
sh:property [ sh:path :providesoutputData ; sh:minCount 1 ; sh:maxCount 1 ; sh:in ( :LevelData ) ; ] ; .

`:LevelDataShape
sh:targetClass :LevelData ;
sh:property [ sh:minInclusive 0 ; sh:maxInclusive 1000 ; sh:datatype xsd:integer ] .
```json
{
    "@context": [],
    "@type": ["Thing"],
    "name": "MyUltrasonicSensor",
    "base": "coap://w3cwot.net:5689/",
    "interactions": [{
        "@type": ["Property"],
        "name": "mylevel",
        "schema": {"type": "number"},
        "writable": false,
        "observable": false,
        "form": [{
            "href": "level",
            "mediaType": "application/json"
        }]
    }]
}
```
Generated Thing Description

W3C WoT Thing Description

```json
{
    "@context": [ ],
    "@type": ["Thing", "iot:LevelCapability"],
    "name": "MyUltrasonicSensor ",
    "base": "coap://w3cwot.net:5689/",
    "interactions": [ {
        "@type": ["Property",
                   "iot:CurrentLevel"],
        "name": "mylevel",
        "schema": { "type": "number"},
        "@type": "iot:LevelData",
        "schema:unitKind": "iot:Centimetre",
        "writable": false,
        "observable": false,
        "form": [ {
            "href": "level",
            "mediaType": "application/json"
        } ]
    } ]
}
```

- [iot.schema.org](https://iot.schema.org)
- [Capabilities](#)
- [Shapes](#)
- [Proposal for shape constraints](#)
- [https://github.com/iot-schema-collab/iotschema/tree/master/shapes](https://github.com/iot-schema-collab/iotschema/tree/master/shapes)
- Code available at: [https://github.com/aparnasai/iotschema/tree/iotschema-TDGenerator/SHACL](https://github.com/aparnasai/iotschema/tree/iotschema-TDGenerator/SHACL)
  branch: iotschema-TDGenerator
Overview

TD RECIPES
Recipes

How to easily:
• Create WoT mash up applications
• Discover, share, and extend mash up specifications
• Discover TDs for your application
• Generate a script to implement an application

How to use Recipes:
1. Discover a Recipe for your mesh-up
2. Discover TDs to implement it
3. Generate a script code
4. Implement application logic

All tasks except the last one are automated
Demo: OverflowProtection on FESTO WS

**Recipe:** If the overflow in the tank occurs, then drain the liquid out.
Siemens work station accessed over Oracle Cloud.

**thingschema.org:** Semantic annotations

**Automated Discovery:** Float Sensor Pneumatic Valve

**Automated Script Code Generation:** script generated and implemented with application logic.
Work Plan

We will provide:

- Restructuring of iotschema.org based on shape constraints
- The online version of Thing Description generator
- Demonstration of applicability of iotschema.org in other models, e.g. IPSO Smart Objects, Amazon IoT, TD from Mozilla and EVRYTHNG etc.
- Extension of home & building domains, and expansion to other domains, e.g., automotive, mobility
Available Resources for iot.schema.org

- Current schema location: http://iotschema.org/
- A stable version should be moved to: http://iot.schema.org/
- GitHub repo: https://github.com/iot-schema-collab
- Working document and notes: https://docs.google.com/document/d/1p8KlUEcQYseoPzvjtkfvVCNAXx3OfyuTUEReB3H2B5M/edit
Thank You!

Questions please...