IoT Extensions for schema.org

Community Teleconference

April 18, 2019

Agenda

- Updates
- schema.org integration proposal
- One Data Model Update – Simple Definition Format
- AOB

Brief Updates

- Community Group
  - Need to elect chairs and create the mail list
- Other updates?

iot.schema.org

iot.schema.org Model - Revisited

Darko Anicic

Aparna Thuluva

Review of the Roadmap for 2019

Among other tasks:

- Review the current iot.schema.org model
- Navigation for Capabilities on our prototype web site: iotschema.org

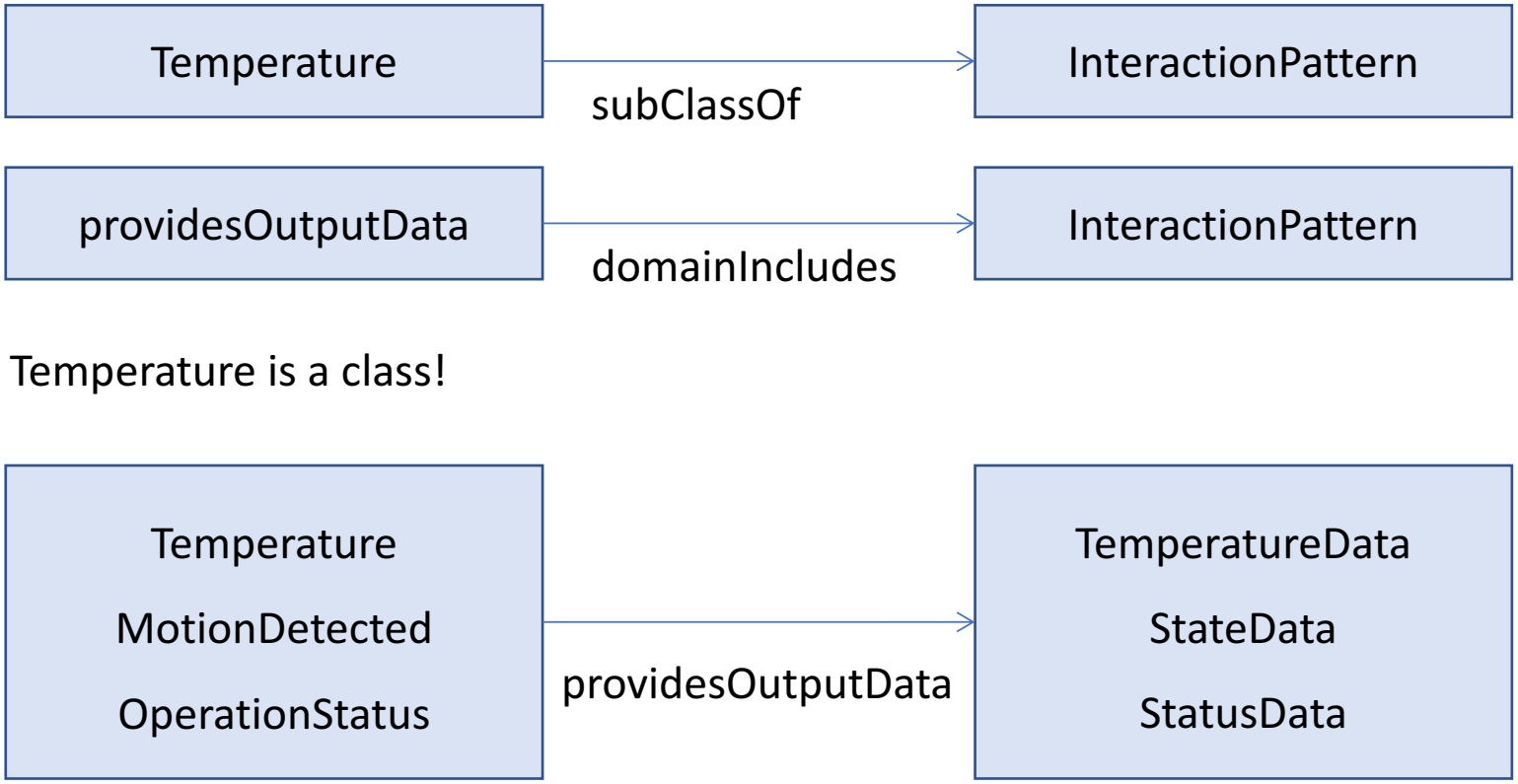
Motivation for changes

iot.schema.org Model

Avoid Mixing Classes and Instances

See Issue #2

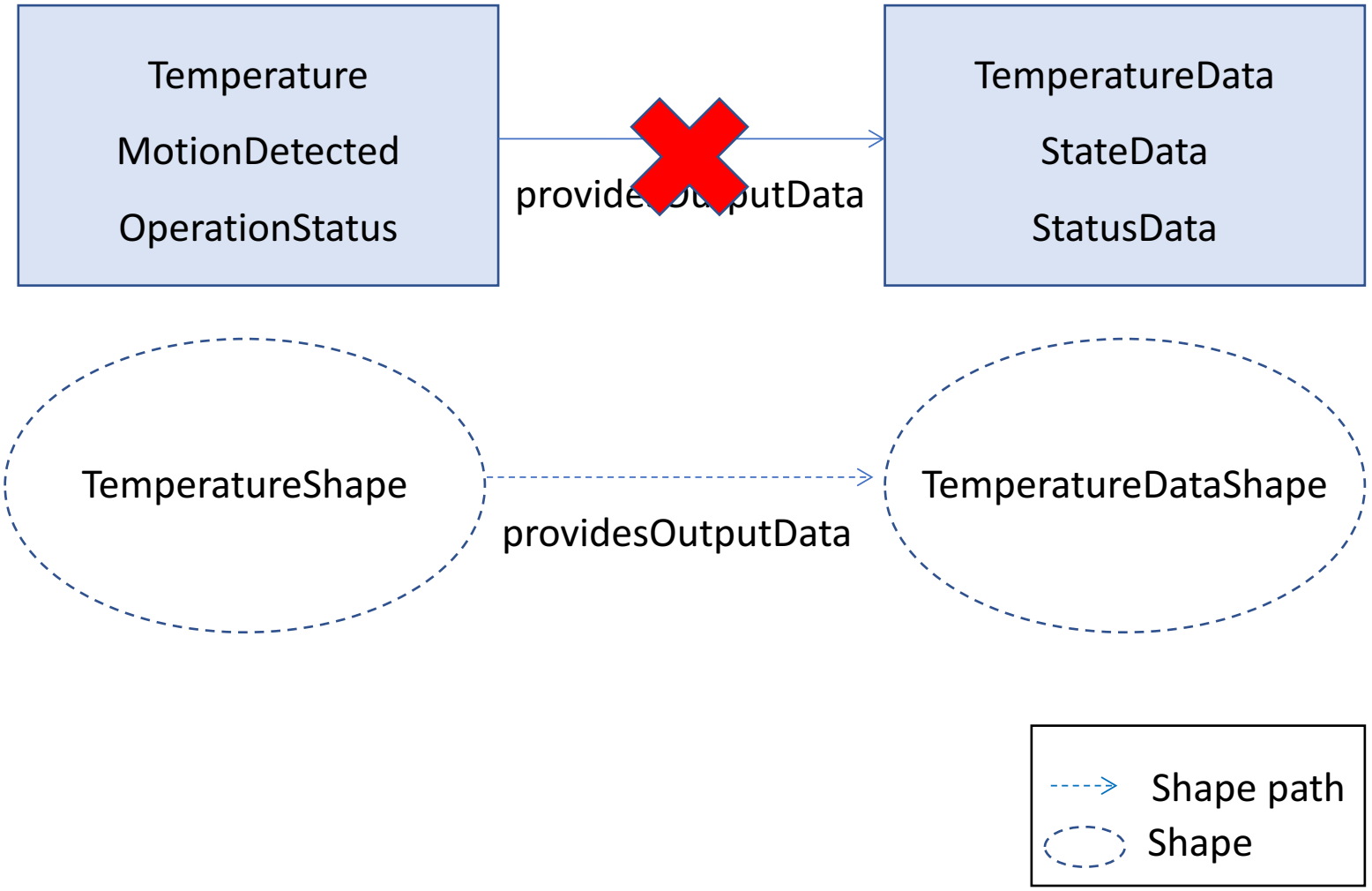
iot.schema.org Example:



Temperature is a class!

Temperature appears to be an instance!- which is not correct

Proposal: Using RDF Shapes in iot.schema.org



Incomplete Content of Schema

AirConditioner

Canonical URL: http://iotschema.org/AirConditioner

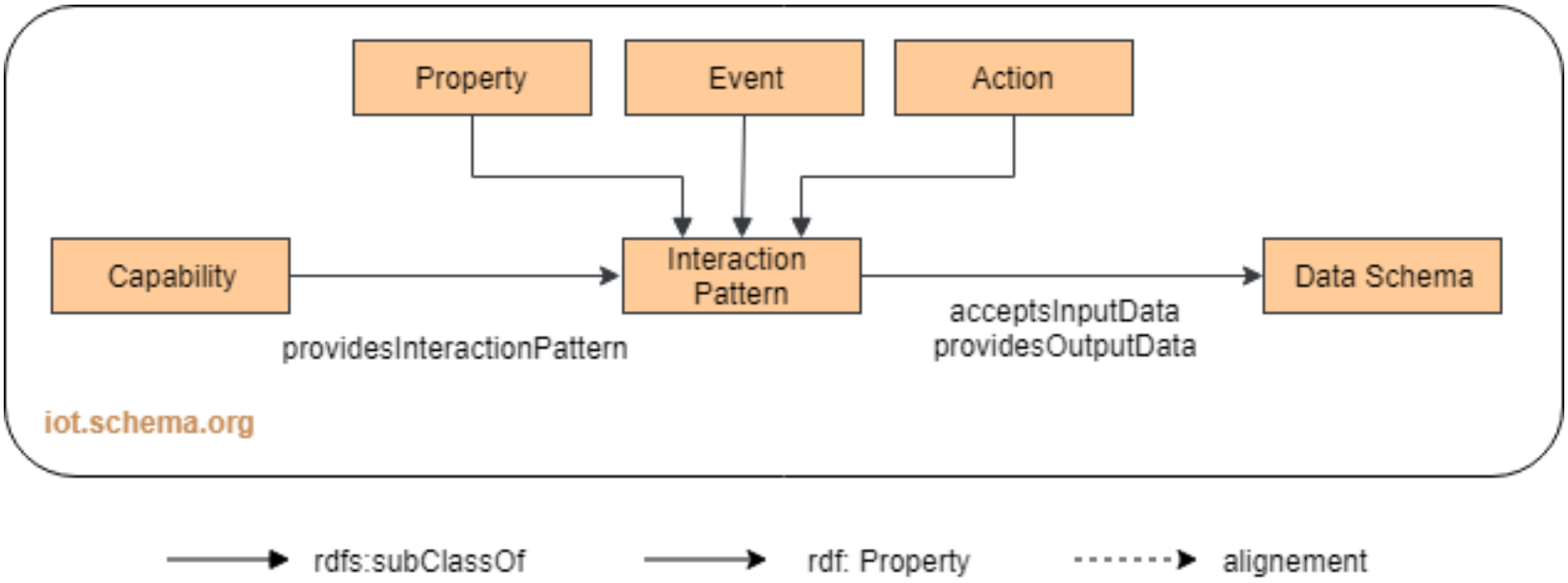
Capability > AirConditioner

A capability for air conditioner

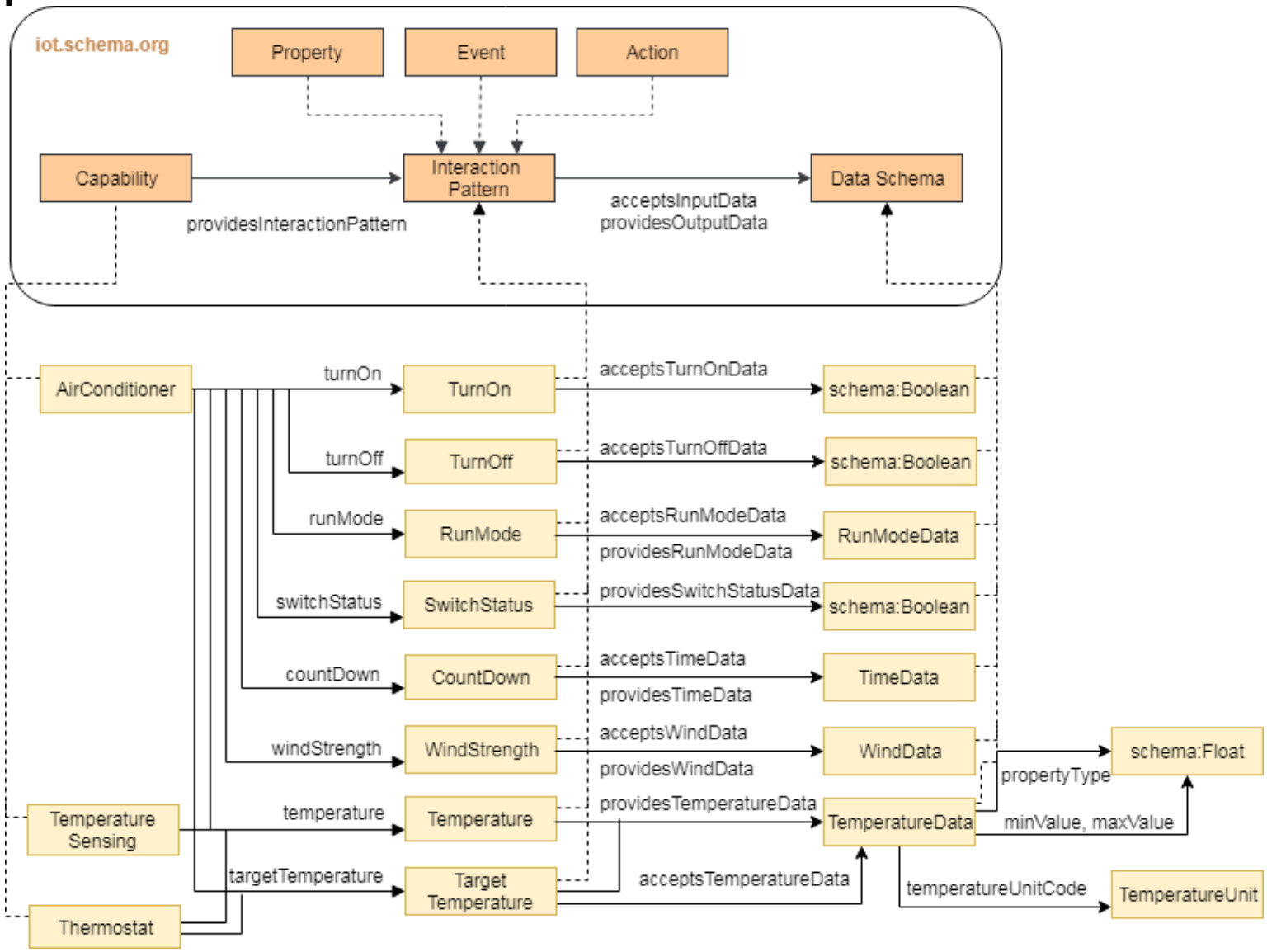
Updated Model

iot.schema.org

Current Capability Model
iot.schema.org



Updated Model



Next Steps

- Review the updated model
- Apply changes to the current schema
- Update our prototype web site: iotschema.org
- Review the iotschema SHACL Shapes

Thank You!

Questions please…

One Data Model

- Harmonize device models across industry
- High level semantic model that is aligned with the pattern we have been using for the prototype definitions
- "Objects" with Event, Action, Property classes
- Data class without any directionality
- Objects compose into things and products

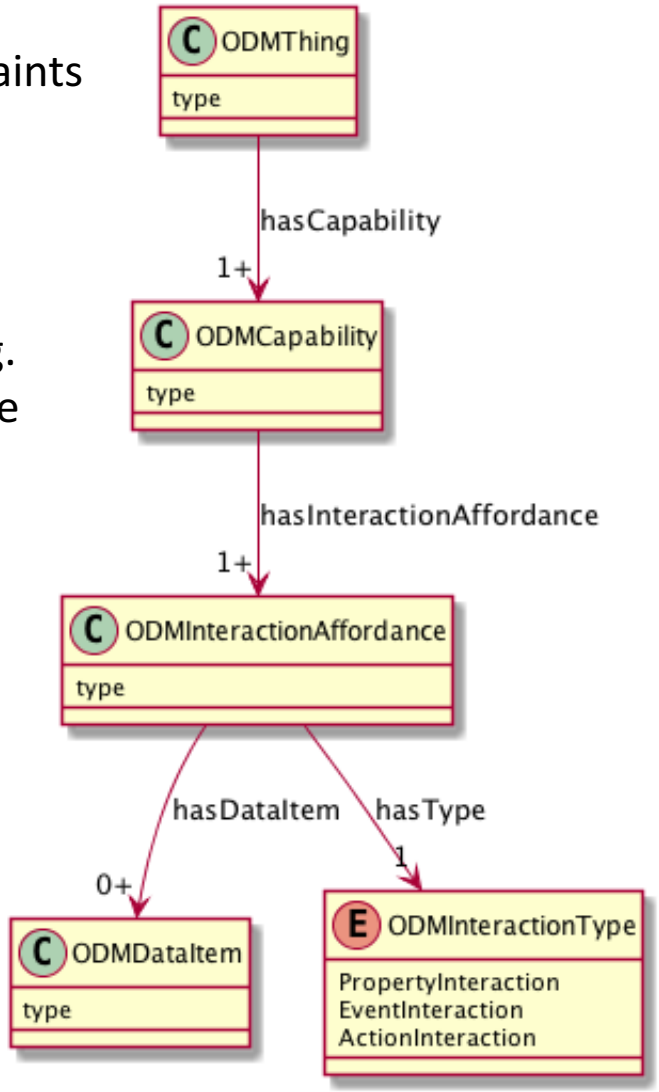
UML Model

Device Type and Constraints
e.g. Thermostat, Light

Composable Objects e.g.
onoff, level, temperature

Events, Actions, Properties

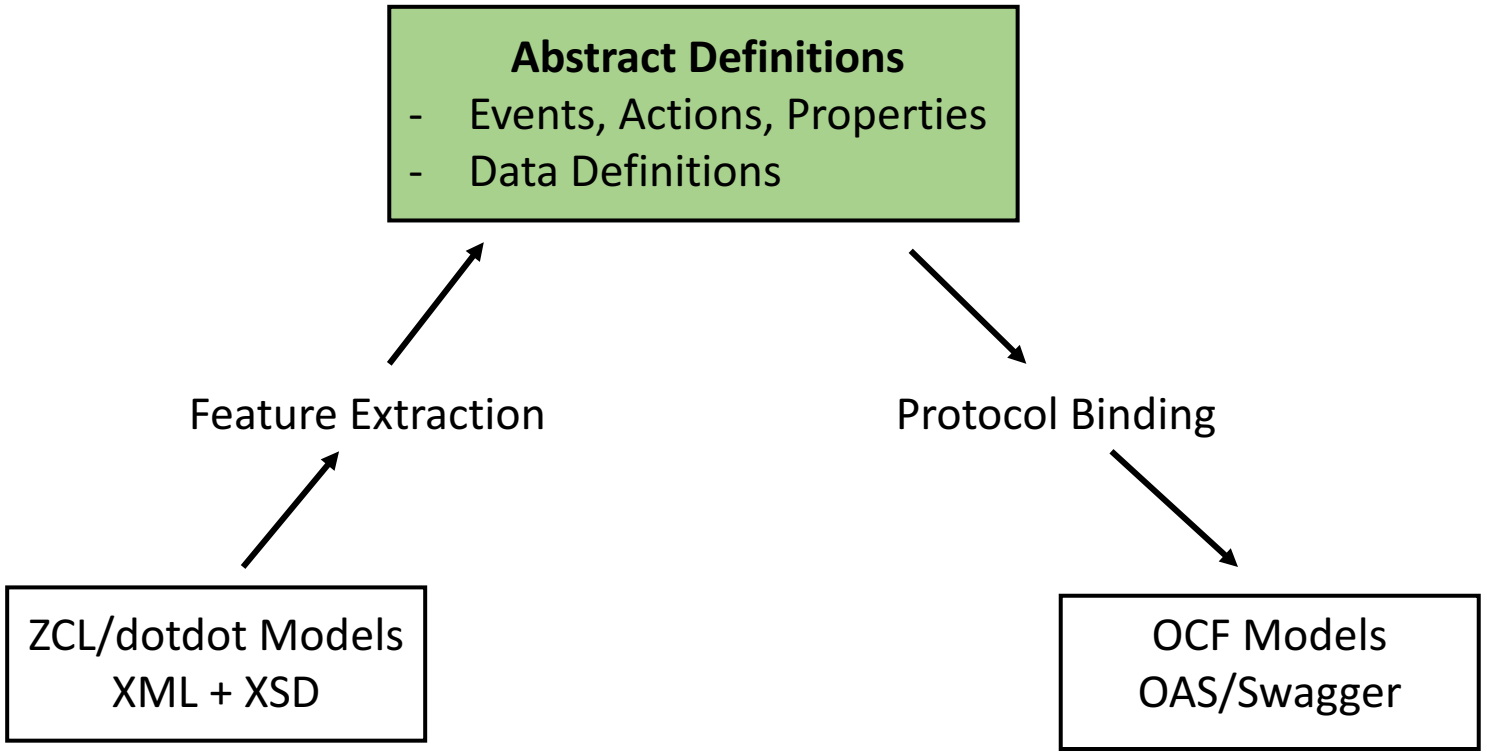
Data Types, Enums



Problems to solve

- Process is to select the "best" from a wide variety of existing models the various orgs have built
- Need to represent the abstract concepts from many design sources
- High level representation that domain experts can use without getting bogged down in the tools
- Methodology and tools
- Simple Definition Format

Support This Pattern



Simple Definition Format

- JSON based DSL for making definitions
- Distill the text down to high information content
- Focus on definitions of objects with events, actions, and properties with semantic data types
- Provide for extensions and constraints that can be applied earlier or later in the life cycle

SDF Example https://github.com/mjkoster/ODM-Examples/SDF2.json

```
{
  "info": {
    "title": "Example file for ODM Simple JSON Definition Format",
    "version": "20190404",
    "copyright": "Copyright 2019 Example Corp. All rights reserved.",
    "license": "http://example.com/license"
  },
  "namespace": {
    "st": "http://smartthings.example.com/capability/odm"
  },
  "defaultNamespace": "st",
  "object": {
    "Switch": {}
  },
  "property": {
    "Switch.value": {
      "type": "string",
      "enum": ["on", "off"]
    }
  },
  "action": {
    "Switch.on": {},
    "Switch.off": {}
  }
}
```

Simple example – Header Part

keywords

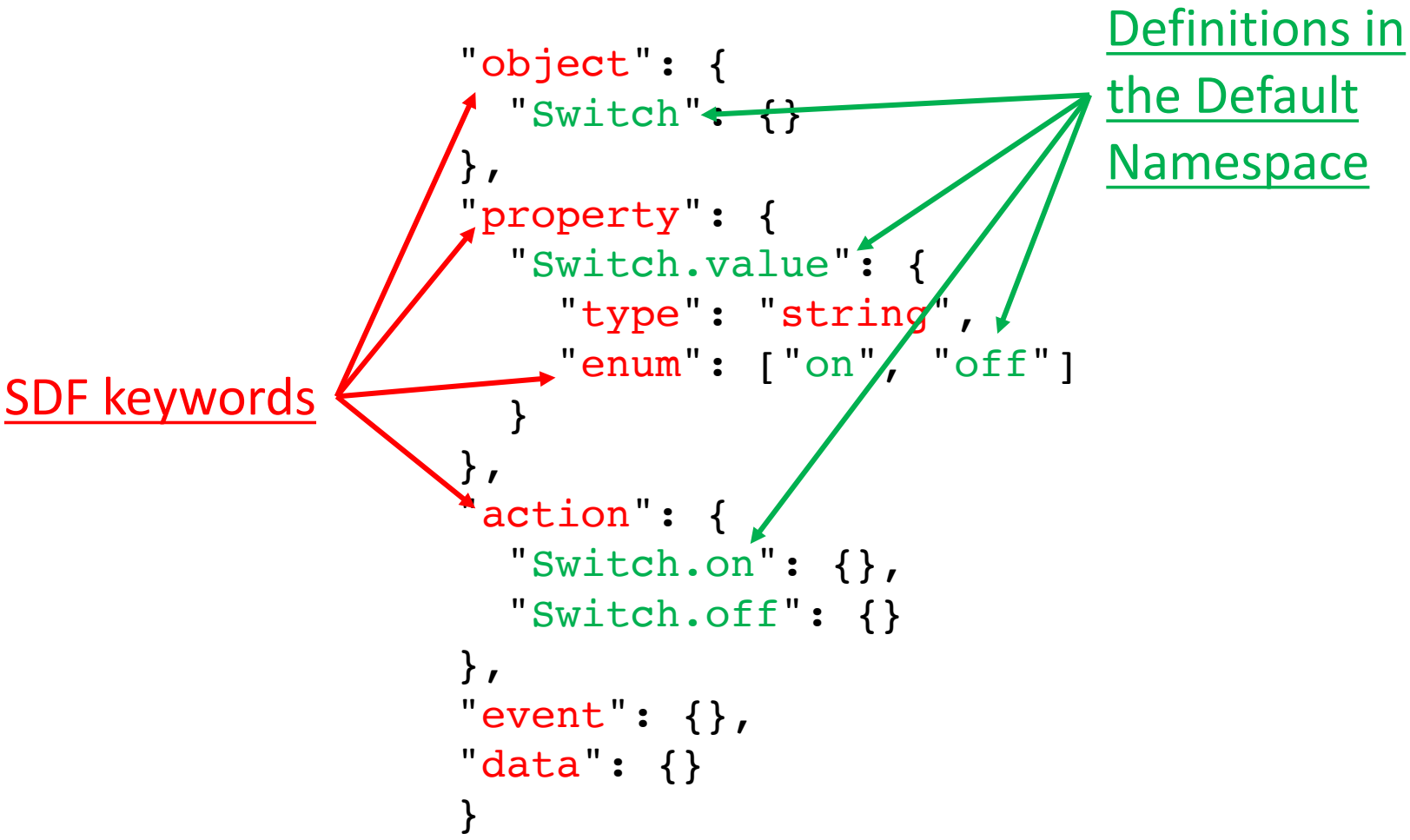
File Information

```
"info": {
  "title": "Example file for ODM Simple JSON Definition Format",
  "version": "20190404",
  "copyright": "Copyright 2019 Xcorp, Inc. All rights reserved.",
  "license": "http://example.com/license"
},
```

curies resolved

```
"namespace": {
  "ocf": "http://openconnectivity.example.org/vocab/odm",
  "st": "http://smartthings.example.com/capability/odm"
},
"defaultNamespace": "st",
```

Definitions



Definitions

- A definition consists of a defined term and a map of it's defined qualities

```
"Switch.value": {
  "type": "string",
  "enum": ["on", "off"]
}
```

Identifier resolution precedence

1 - Explicit namespace prefix (e.g. "st:Switch")

2 - SDF Keywords:

https://github.com/mjkoster/ODM-Examples/blob/master/SDF2-Schema.json

3 – defaultNamespace

4 – Context of the definition (Object), local identifiers

SDF top level Keywords

- info
  - title, version, copyright, license
- namespace, defaultNamespace
- object, property, action, event, data (definitions)

Object qualities + common qualities

```
"description": {
  "type": "string"
},
"title": {
  "type": "string"
},
"id": {
  "type": "string"
},
"name": {
  "type": "string"
},
"type": {
  "type": "string"
},
```

```
"optional": {
  "type": boolean"
{
"extends": {
  "type": "string"
},
"refines": {
  "type": "string"
},
"includes": {
  "type": "string"
},
```

Property Qualities

- All of the Object Qualities
- All of the Data Qualities

```
"units": {
  "type": "string"
},
"scaleMinimum": {
  "type": "number"
},
"scaleMaximum": {
  "type": "number"
},
"observable": {
  "type": "boolean"
},
```

```
"nullable": {
  "type": "boolean"
},
"encoding": {
  "type": "object",
  "properties": {
    "widthInBits": {
    "type": "number"
    }
  }
},
"contentFormat": {
  "type": "string"
}
```

Data Qualities (JSON Schema)

```
"type": {
  "type": "string",
  "enum": [ "number", "string",
"boolean", "integer", "array",
"object" ]
  },
"enum": {
  "type": "array"
},
"const": {
  "type": { "oneOf" : [
"number", "string", "boolean",
"array", "object", "null" ] }
},
"default": {
  "type": { "oneOf" : [
"number", "string", "boolean",
"array", "object", "null" ] }
},
```

```
"pattern": {
  "type": "string"
},
"minimum": {
  "type": "number"
},
"maximum": {
"type": "number"
},
"multipleOf": {
"type": "number"
},
"maxLength": {
"type": "number"
},
"minLength": {
"type": "number"
},
```

Structured Data

```
"oneOf": {
  "type": "array",
  "minItems": 1
},
"anyOf": {
  "type": "array",
  "minItems": 1
},
"allOf": {
  "type": "array",
  "minItems": 1
},
"items": {
  "oneOf": [
    { "type": "array" },
    { "type": "object" }
  ]
},
```

```
"contains": {
  "oneOf": [
    { "type": "array" },
    { "type": "object" }
  ]
},
"minItems": {
  "type": "number"
},
"maxItems": {
  "type": "number"
},
"properties": {
  "type": "object"
},
"readOnly": {
  "type": "boolean"
},
"writeOnly": {
  "type": "boolean"
}
```

SDF Example Definition

https://github.com/mjkoster/ODM-Examples/blob/master/SDF2-SwitchLevel.json

```
"object": {
  "SwitchLevel": {}
},
"property": {
  "SwitchLevel.level": {
    "type": "SwitchLevel.levelData",
    "readOnly": false
}
},
"action": {
  "SwitchLevel.setLevel:": {
    "data": [
      "SwitchLevel.levelData",
      "SwitchLevel.rateData"
    ]
  }
},
```

```
"data": {
  "SwitchLevel.levelData": {
    "type": "number",
    "minimum": 0,
    "maximum": 100,
    "multipleOf": 1
  },
  "SwitchLevel.rateData": {
    "type": "number",
    "minimum": 0,
    "maximum": 65535,
    "multipleOf": 1
  }
}
```

Example JSON-LD Result

```
{
  "@id": "st:SwitchLevel.level",
  "rdfs:comment": "The current level setting",
  "rdfs:label": "SwitchLevel level Property",
  "@type": "odm:PropertyInteraction",
  "rdfs:subClassOf": "odm:InteractionAffordance",
  "odm:hasDataItem": "st:SwitchLevel.levelData"
},
{
  "@id": "st:SwitchLevel.setLevel",
  "rdfs:comment": "Action to set the level",
  "rdfs:label": "SwitchLevel setLevelAction",
  "@type": "odm:ActionInteraction",
  "rdfs:subClassOf":"odm:InteractionAffordance",
  "odm:hasDataItem": [
    "st:SwitchLevel.levelData"
    "st:SwitchLevel.rateData"
  ]
}
```

Opportunity

- Share the definition format and tools between schema.org IoT extensions and One Data Model
- Provide domain expert friendly tools for creating and managing definitions
- ODM sourced definitions can be processed into schema.org extensions
- Schema.org IoT sourced definitions can use SDF tools and can become ODM definitions
- Models and definitions can be created within the domain expert venues

Other Business?

- AOB
- Adjourn