iot.schema.org

August 23, 2017

Agenda

- Status (Michael)
  - Report out WoT F2F and WISHI (July)
  - Roadmap through November
  - Strawman Definitions
- Design patterns and tools (Darko)
  - Capability definitions, shapes constraints
  - Authoring in markdown
- Next steps (All)

W3C WoT Meeting Duesseldorf

- Expect to use iot.schema.org definitions for the November Plugfest
- Prototype of semantic annotation in web applications using WoT Thing Description
  - Discovery using Semantic Annotation
  - Integration into applications
- Provide a set of definitions for common things, following the current design pattern

IRTF T2TRG WISHI Workshop

- Semantic Interoperability and Hypermedia
  - https://github.com/t2trg/2017-07-wishi
- In conjunction with IETF99
- Good discussion around iot.schema approach
- Potential collaboration: Fairhair, oneM2M, IPSO

Roadmap through November

- Activate the W3C WoT Community Group
  - Define extension to the charter
- Strawman Definitions
  - End of August
- HTML documentation using schema.org
- Hosting on schema.org experimental area
- WoT Thing Description prototypes

For November WoT PlugFest

- Create a set of Modular, Reusable, and Interoperable Capability definitions for some common interactions.
- Host the definition files
- Host HTML documentation
- Provide example TDs that are annotated with the Capability definitions.

List of Capability Definitions

- General
  - temperature
  - illuminance
  - motion/occupancy
  - smoke detector
  - co, co2 detector
  - contact sensor
  - shade control
  - energy used
- Light
  - onoff switch
  - level/brightness
  - rgb color
  - hs color
  - xy color
  - color temperature
- Thermostat
  - temperature
  - setpoint
  - heating setpoint
  - cooling setpoint
  - operating state
  - thermostat mode
  - fan mode

Design Pattern Update

Review and Backup

Overview of iot.schema.org

- High level semantic interoperability layer
- Interoperable definitions for connected things
- Protocol-neutral abstractions for affordances exposed by connected things
- Venue to create and manage definitions for multiple vertical application domains
- Design pattern compatibility with WoT Thing Description Classes: Event, Action, Property

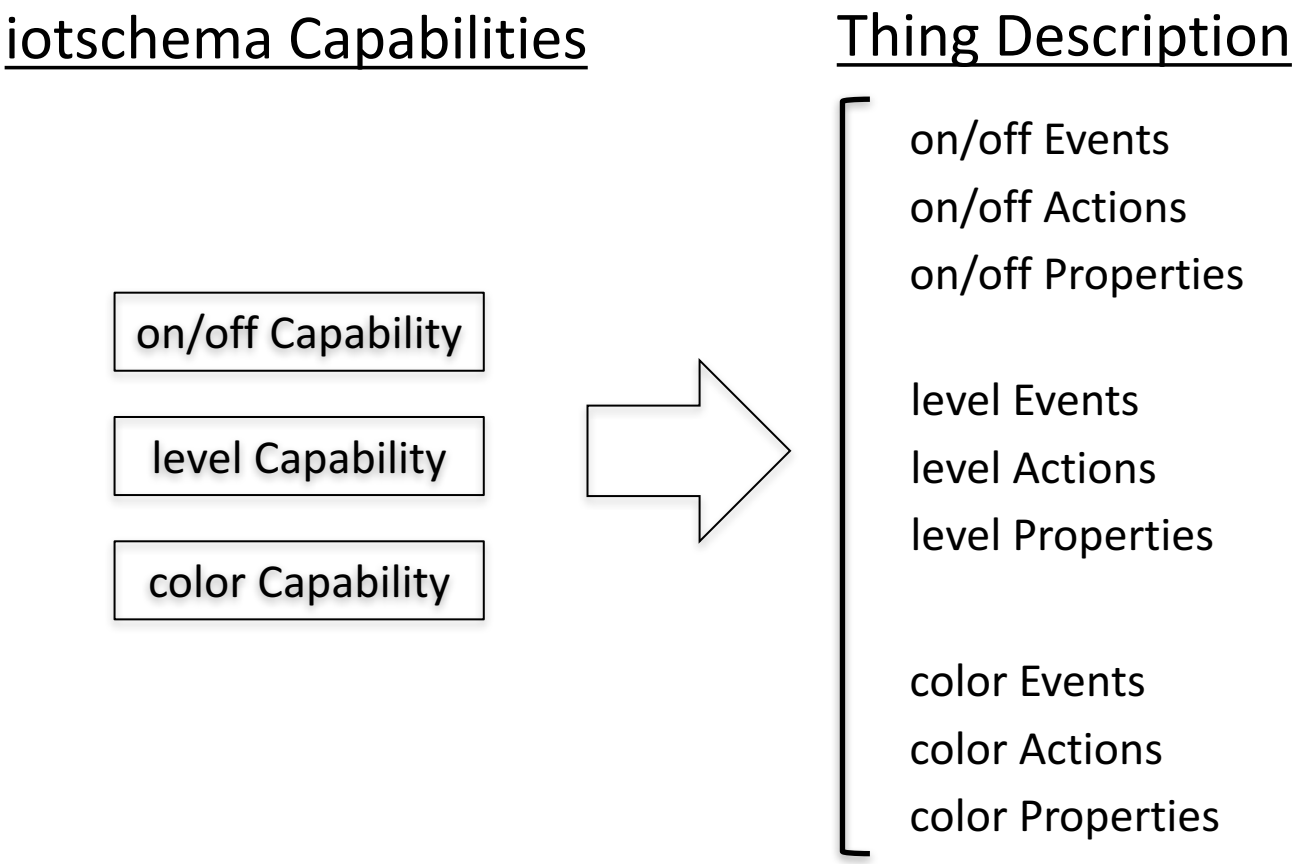
Capability Abstraction

- Common abstraction => "Capability" to make definitions more modular
- 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**

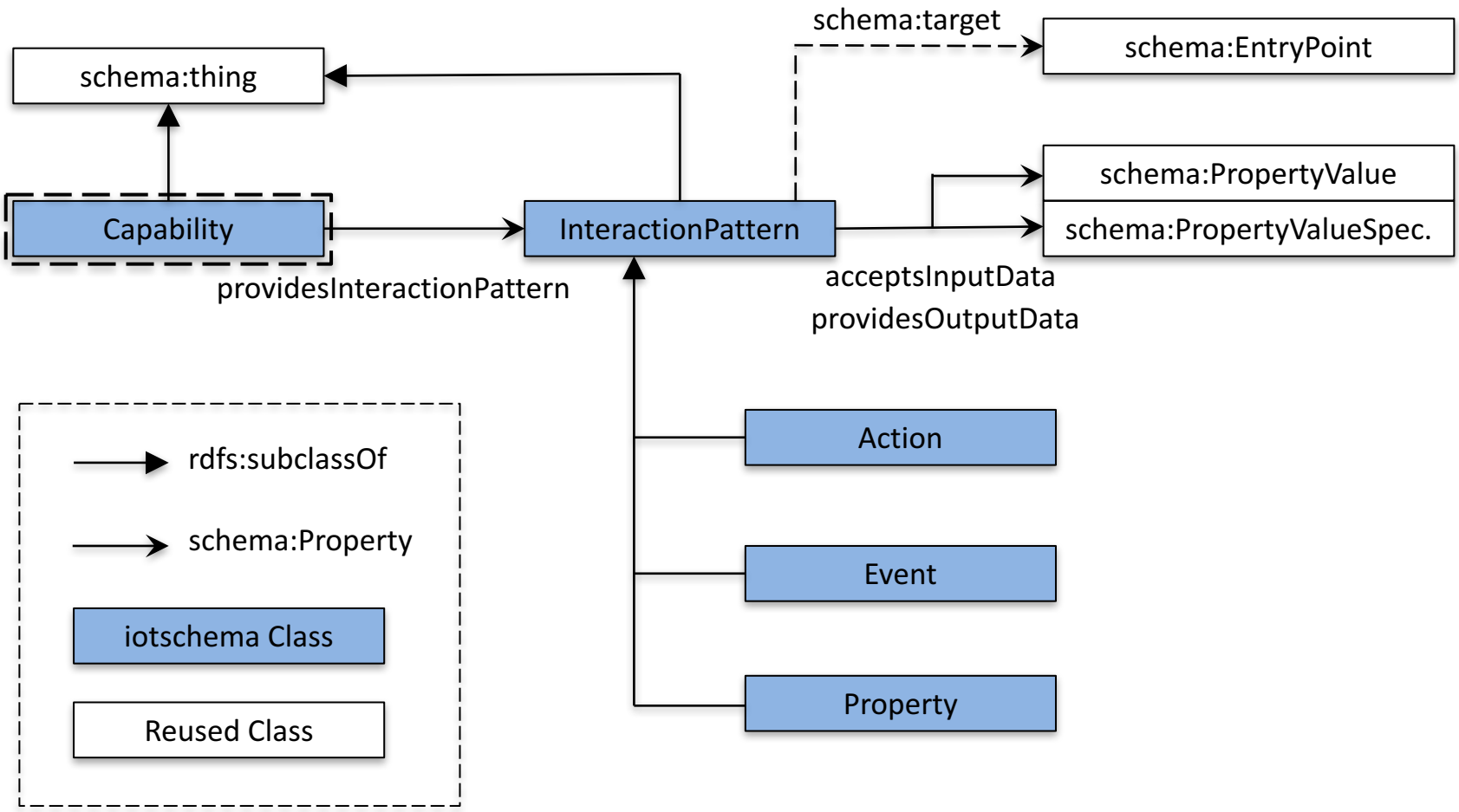
Capability Abstraction (2)

- Capabilities are composed together to define the functional interface of a connected thing
- For example, a connected light bulb may have
  - On/off switch capability
  - Dimming/Brightness capability
  - Color Control capability
  - Energy Monitoring capability
- A TD for this light bulb would contain events, actions, and properties of all capabilities

Capability Composition



iotschema Capability Pattern



Capability Composition

- Capabilities can be sub-things or can add to events, actions, and properties in a flat TD
- Layered semantic annotation
  - @type: [thing, light]
  - capability: [onoff, level, color]

Example iotschema Definition

```
"@id": "LevelCapability",
"@type": "rdfs:Class",
"rdfs:subClassOf": { "@id": "Capability" },
"rdfs:comment": "Level Sensing and Control Capability",
"rdfs:label": "Level Capability",
"domain": {"@id": "Common"},
"providesInterctionPattern": [
  {
    "@id": "CurrentLevel",
    "@type": "rdfs:Class",
    "rdfs:subClassOf": { "@id": "Property" }
    "acceptsInputData": {
      "@id": "LevelData",
      "domain": { "@id": "Common" },
      "rdfs:comment": "Level data",
      "rdfs:label": "Level Data",
      "rdfs:subClassOf": { "@id": "schema:PropertyValue" },
      "schema:propertyType": { "@id": "schema:Number" },
      "schema:unitCode": { "@id": "qudt:Percent" }
      "rdfs:comment": "Current level property",
      "rdfs:label": "Current Level"
    },
    (providesOutputData)
```

Example Definition (cont'd)

```
{
  "@id": "TransitionTime",
  "@type": "rdfs:Class",
  "rdfs:subClassOf": { "@id": "Property" }
  "acceptsInputData": {
    "@id": "TransitionTimeData",
    "domain": { "@id": "Common" },
    "rdfs:comment": "Transition time",
    "rdfs:label": "Transition Time",
    "rdfs:subClassOf": { "@id": "schema:PropertyValue" },
    "schema:propertyType": { "@id": "schema:Number" },
    "schema:unitCode": { "@id": "qudt:SecondTime" }
  },
  "providesOutputData": {
    "@id": "TransitionTimeData"},
    "domain": { "@id": "Common" },
    "rdfs:comment": "Transition time",
    "rdfs:label": "Transition Time",
    "rdfs:subClassOf": { "@id": "schema:PropertyValue" }
    "schema:propertyType": { "@id": "schema:Number" },
    "schema:unitCode": { "@id": "qudt:SecondTime" }
  }
},
```

Example Definition (cont'd)

```
    {
      "@id": "SetLevel",
      "@type": "rdfs:Class",
      "rdfs:subClassOf": { "@id": "common:Action" },
      "rdfs:comment": "Set level Action",
      "rdfs:label": "Set Level Action",
      "domain": { "@id": "Common" },
      "acceptsInputData": [
          { "@id": "TransitionTimeData" },
          { "@id": "LevelData" }
        ]
    }
  ]
}
```

Example Instance – TD Interaction

```
{
  "@context": [
    "http://w3c.github.io/wot/w3c-wot-td-context.jsonld",
    "http://w3c.github.io/wot/w3c-wot-common-context.jsonld",
    {"iotschema": "http://iot.schema.org/common#"}],
  "@type": [ "Thing", "BuildingSpace" ],
  "name": "Lamp",
  "capability" : "iotschema:LevelCapability",
  "interactions": [{
    "name": "Set brightness level",
    "@type": ["Action", "iotschema:SetLevel"],
    "inputdata":  {
      "type": "object",
      "properties": {
        "level": {
          "type": ["iotschema:LevelData", "number"]},
        "time": {"type": ["iotschema:TransitionTimeData", "number"]}
    }},
    "link": [{
      "href": "/example/light/setbrightness",
      "mediatype": "application/json"
      }]
    },
```

Example Instance – TD Interaction (2)

```
{
    "name": "Current brightness level",
    "@type": ["Property", "iotschema:CurrentLevel"],
    "inputdata":  { "type": ["iotschema:LevelData", "number"] },
    "outputdata": { "type": ["iotschema:LevelData", "number"] },
    "link": [
      {
       "href": "/example/light/currentbrightness",
       "mediatype": "application/json"
      }]},
  {
    "name": "Transition time",
    "@type": ["Property", "iotschema:TransitionTime"],
    "inputdata":{"type":["iotschema:TransitionTimeData","number"] },
    "outputdata":{"type":["iotschema:TransitionTimeData","number"] },
    "link": [
      {
       "href": "/example/light/transitionTime",
       "mediatype": "application/json"
      }]
}]}
```