iot.schema.org

Community Teleconference Agenda

April 19, 2018

Agenda

- Prototype HTML website
- Report out from WISHI plugfest at IETF 101
- Report out from plugfest at W3C WoT Prague
- Feature of Interest
- RDF shape constraints
- Charter update for WoT Community Group
- Next steps
- AOB

HTML Website Prototype

- iotschema.org
- adapted the schema.org HTML tools
- still a little rough but browseable

WISHI Semantic Interop

- WISHI sponsored at IETF 101 Hackathon
- Focus on Semantic Interoperability and Discovery
- Participants from Samsung, Ericsson, Eurecom, Siemens, Acklio
- Used WoT infrastructure, LWM2M, OCF, and vendor specific e.g. Philips Hue
- Used a Thing Directory and WoT Thing Description with semantic annotation from the iot.schema.org prototype definitions

W3C Web of Things Plugfest

- In conjunction with WoT F2F in Prague
- Focus on semantic annotation, protocol binding, and interaction through proxy servients
- Included automotive domain
- Used Thing Directory and WoT Thing Description with semantic annotation from the iot.schema.org prototype definitions

Learning from Plugfests

- Directory-based discovery using semantic annotation and queries is a workable method
- Ease of use for developers is very important
    - Simplified conceptual models for annotation and discovery are helpful – simple semantic categories
    - Ease the learning curve for domain experts creating new definitions
- Breadth of definitions and models is expected
- Feature of Interest modeling is a clear gap

What is new in iot.schema.org

- In preparation to W3C WoT PlugFest -

- The schema contains more terms related to PlugFest devices
- Few Event specifications have been added
- Few Action specifications (for writable Properties) have been added
- Attributes “writable” and “observable” added for Properties

Working Plan

- Conclusions from W3C WoT Meeting -

- Integration of vocabularies from Haystack (home & building)
- Vocabulary for the device management & bootstrapping is needed (consider the OMA LWM2M work)
- Extension to other domains, e.g., automotive, mobility
- Demonstration of applicability in other ecosystems, e.g. Amazon IoT, TD - Mozilla and EVRYTHNG, OMA SpecWorks etc.
- Clear process how to contribute and use iot.schema.org

Feature of Interest Modeling

- End to end semantic integration requires more than capabilities
- Needed to make a logical connection between capabilities and entities in the physical world
  - Door lock capability and "Front Door" entity
  - Also a property of an entity – level control capability of a light acts on the illuminance property of a space
- Modeling "branch" entities in the GENIVI VSS automotive specification requires domain specific definitions for doors, mirrors, seats and their positions on the vehicle

Motivation for Feature Of Interest Pattern

- Binds Capability and Interaction Patterns to real-world objects
- This provides information about the environment in which sensing/actuating is applied
- PlugFest use cases prove that the Feature of Interest (FoI) pattern is needed in iot.schema.org

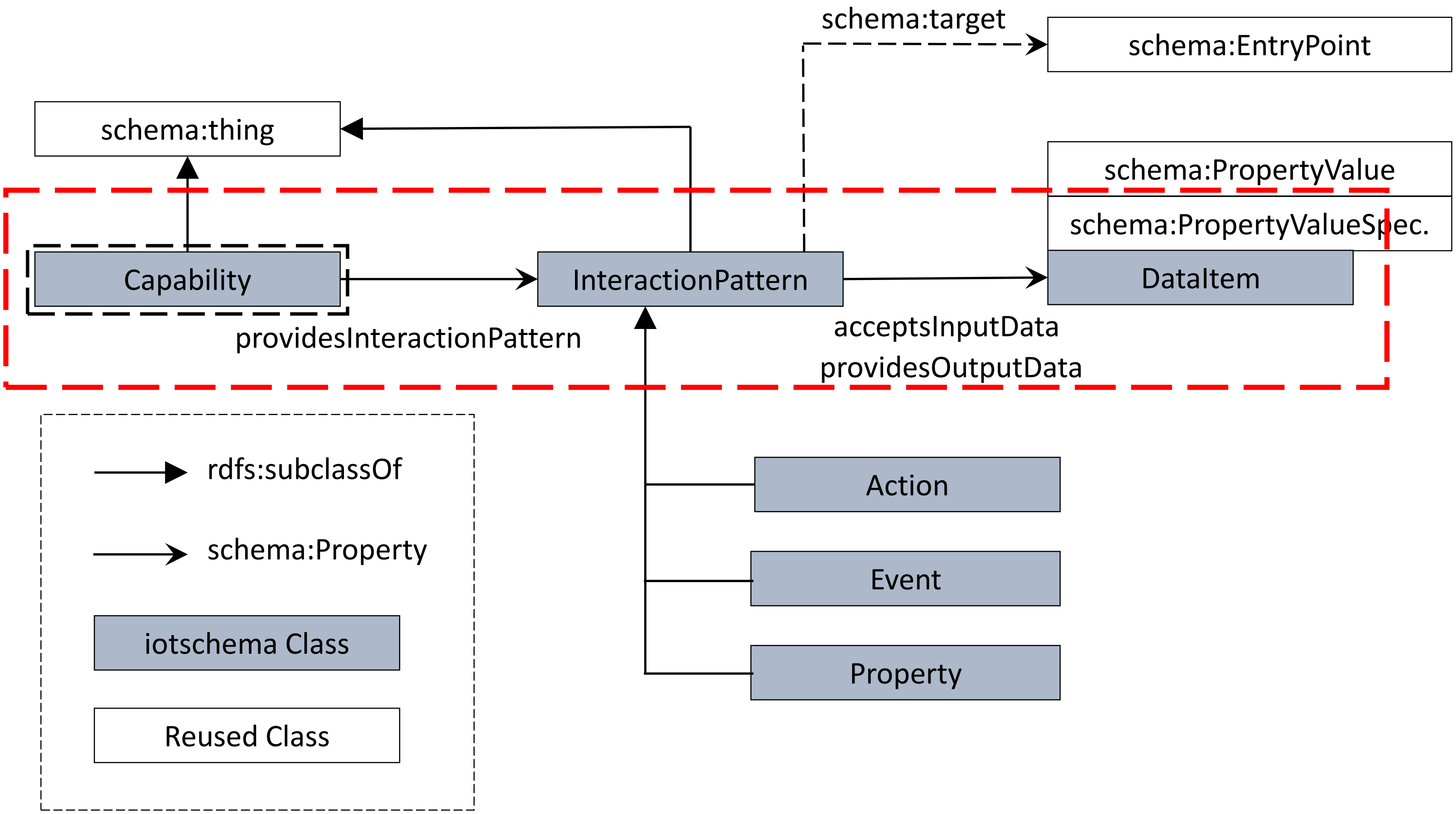
Feature of Interest

- Feature of Interest concept, originally from the O&M ontology
- Features of Interest are entities in the real world that are the target of sensing - https://www.w3.org/2005/Incubator/ssn/wiki/SSN_Skeleton
- Features of Interest have Observable Properties
- Observable Properties are observed by Sensors

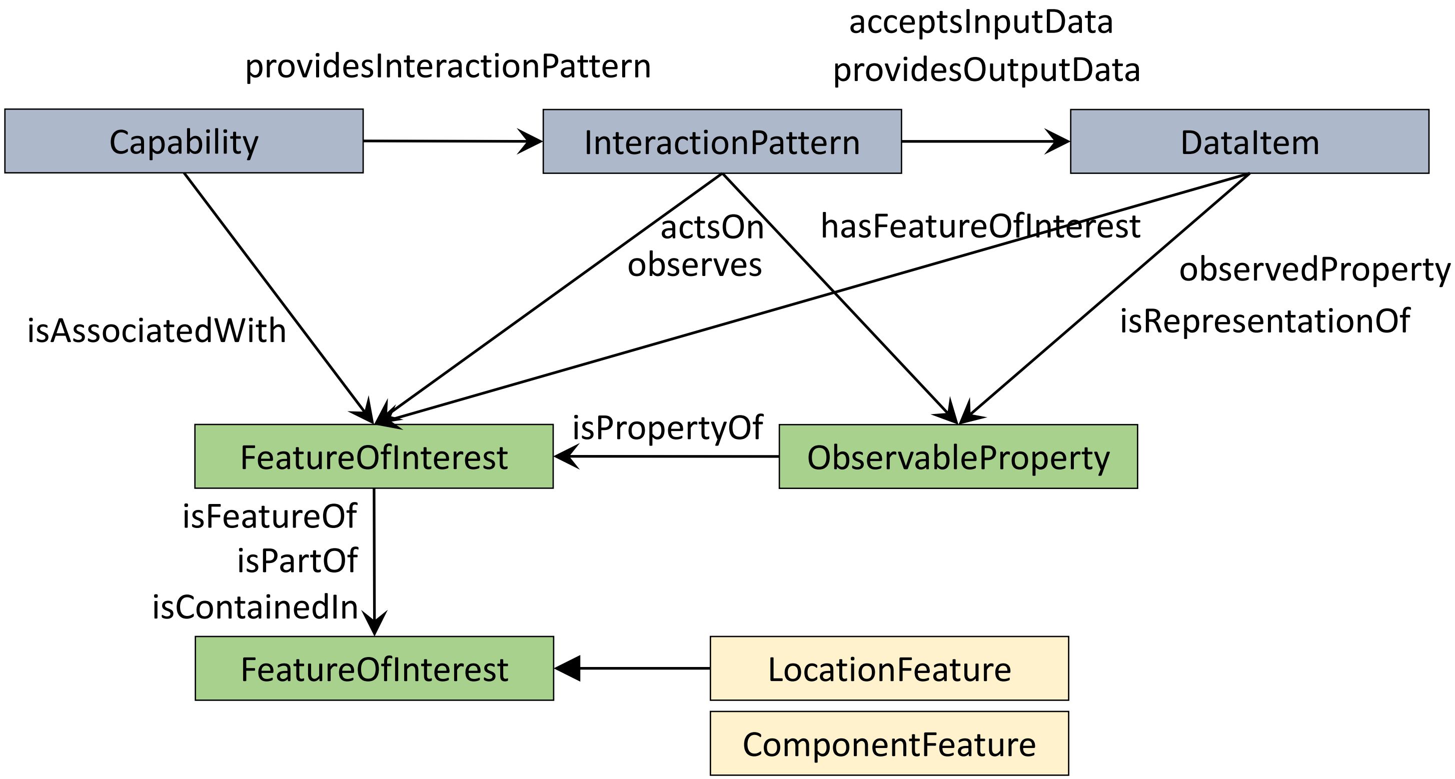
Feature of Interest

- This pattern may be extended for actuators to act on properties of Features of Interest
- Use property types like "actsOn" and "observes" to describe the relationship between Capabilities and Features of Interest
- Also need to relate FoI to other FoI in order to describe structural relationships
  - A water valve is "partOf" a clothes washer
  - More property types can be created to describe domain specific relationships

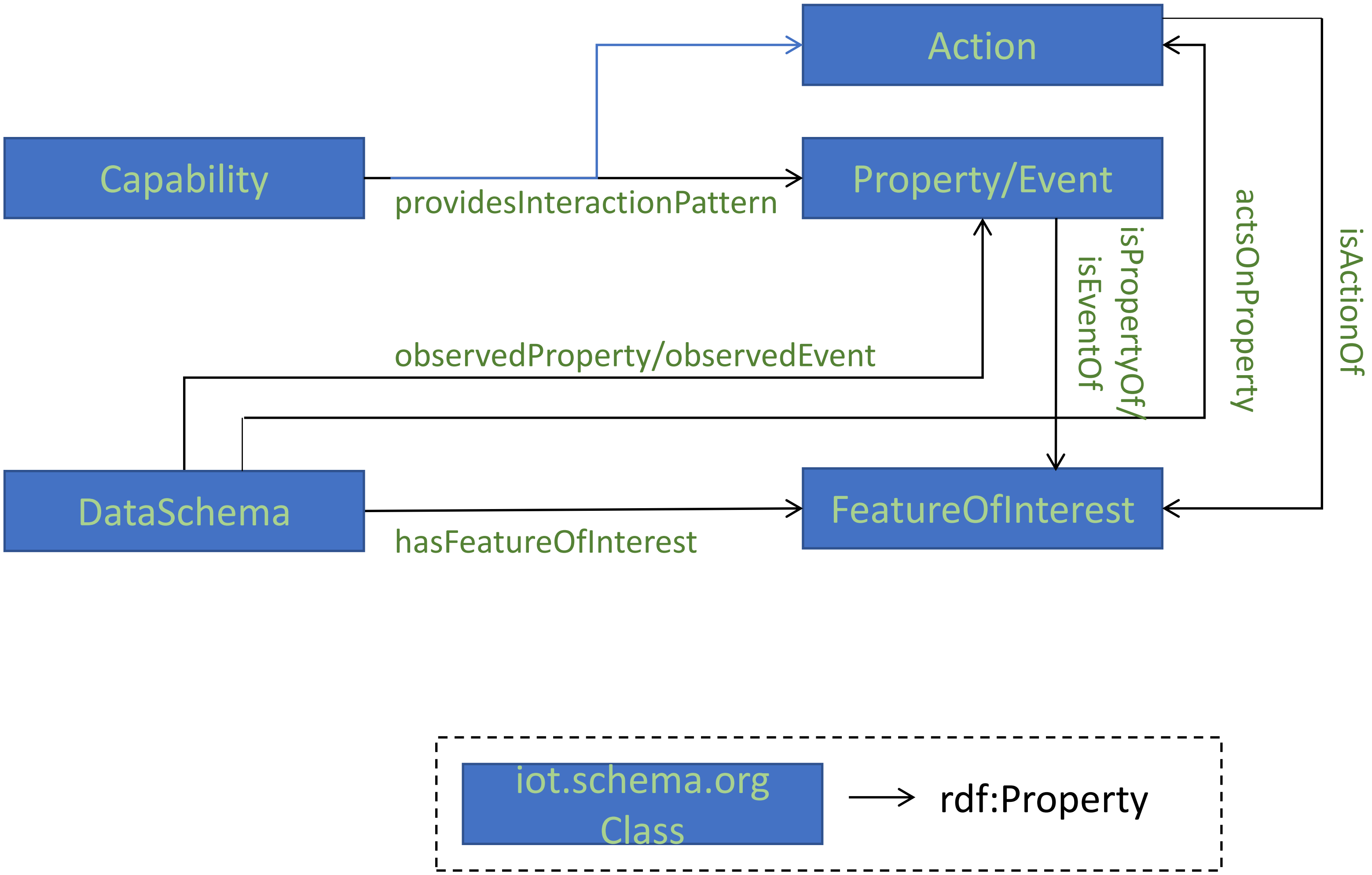
iotschema Capability Pattern



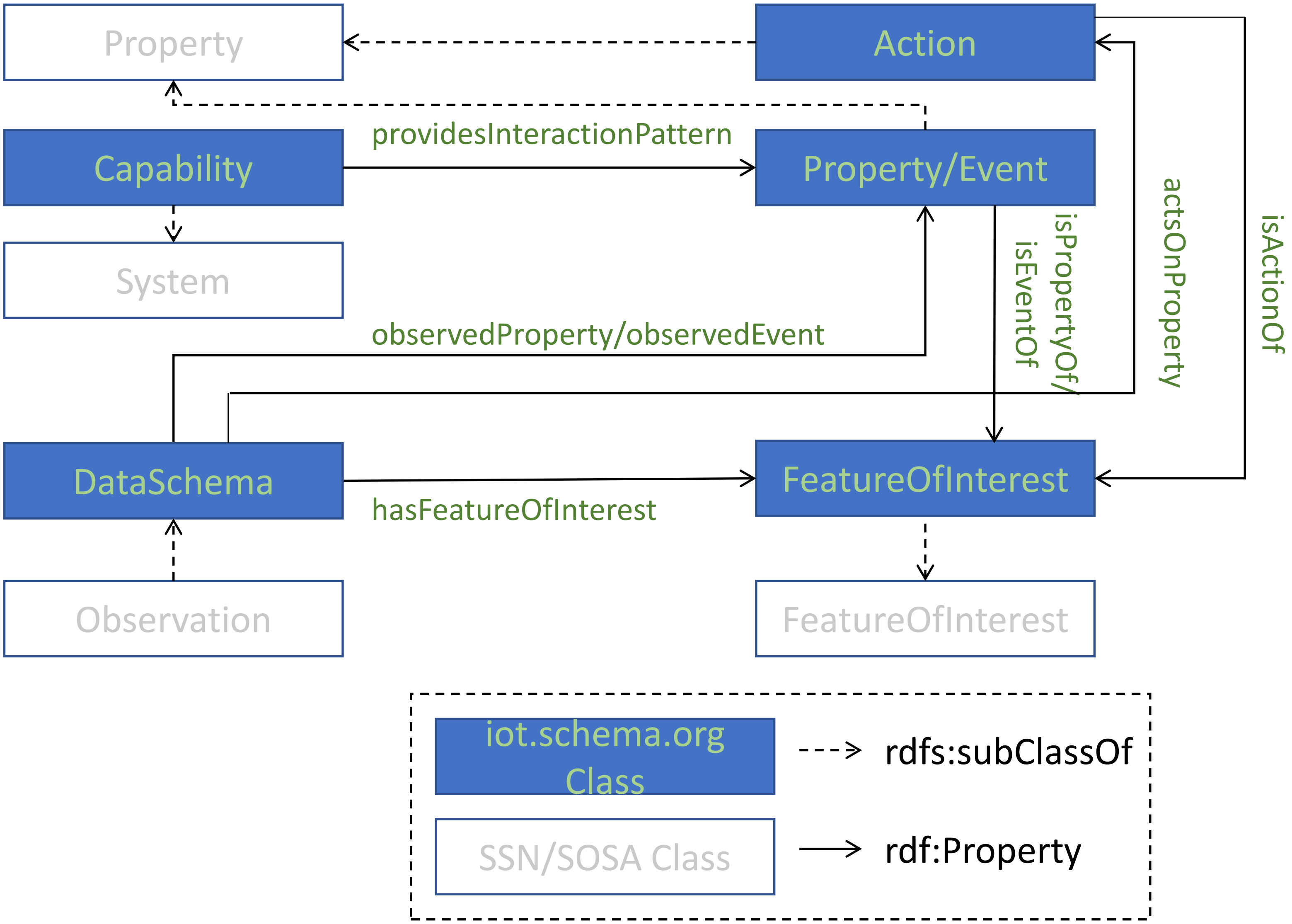
Feature of Interest Integration Patterns



Feature Of Interest Pattern



Feature Of Interest Integration Pattern



Feature of Interest Example

```
   {
              "@id": "iot:TemperatureSensing",
              "rdfs:subClassOf": { "@id": "iot:Capability" },
              "iot:providesInteractionPattern": [          {          "@id": "iot:Temperature“     }]
  }, {
              "@id": "iot:Temperature",
              "rdfs:subClassOf": { "@id": "iot:Property" },
              "iot:isPropertyOf": {"@type": "iot:Room"},
              "iot:providesOutputData": {    "@id": "iot:TemperatureData“  }
}, {
              "@id": "iot:TemperatureData",
              "rdfs:subClassOf": { "@id": "iot:DataSchema" },
              "iot:hasFeatureOfInterest": {"@type": "iot:Room"},
              "iot:observedProperty": "iot:Temperature",
              "schema:propertyType": { "@id": "schema:Float" },
              "schema:unitCode": { "@id": "iot:TemperatureUnit" },
              "schema:minValue": "schema:Float",
              "schema:maxValue": "schema:Float"
}
```

Feature of Interest

- Define a starting set of property types for relations between capabilities and FoI
- Create some definitions for Features of Interest in relevant application domains (smart home, automotive)
- Incorporate FoI and relation types into Thing Directories
    - Add Semantic annotation for FoI on registration
    - Enable semantic filter for FoI on discovery

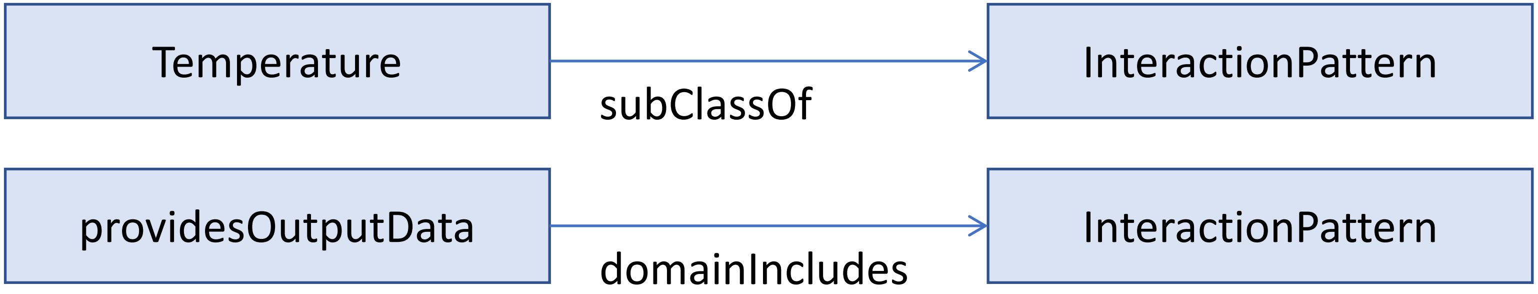
RDF Shape Constraints

- Use RDF Shape Constraints to define data shapes
- Common approaches have disadvantages
  - PropertyValueSpecification (schema.org)
  - JSON Schema

Avoid Mixing Classes and Instances

See Issue #2

iot.schema.org Example:



Temperature

subClassOf

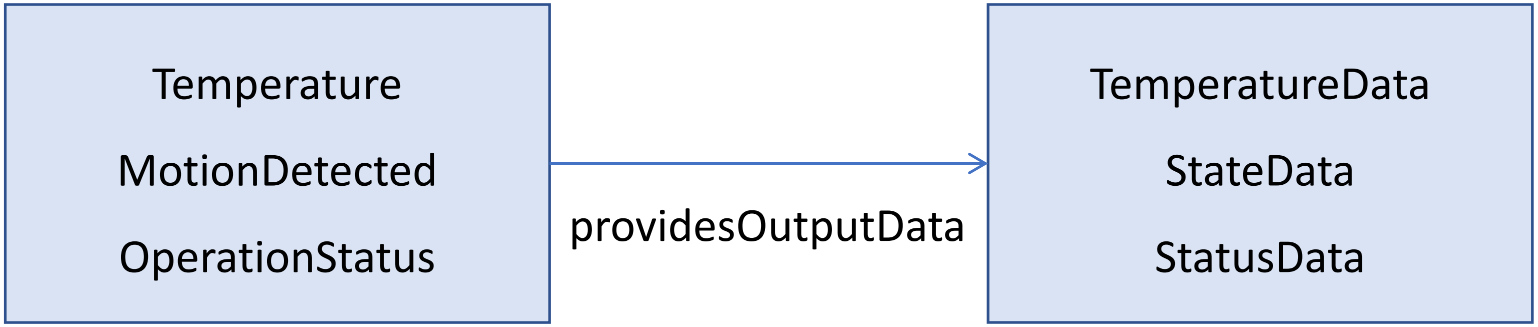
InteractionPattern

providesOutputData

domainIncludes

InteractionPattern

Temperature is a class!



Temperature

MotionDetected

OperationStatus

providesOutputData

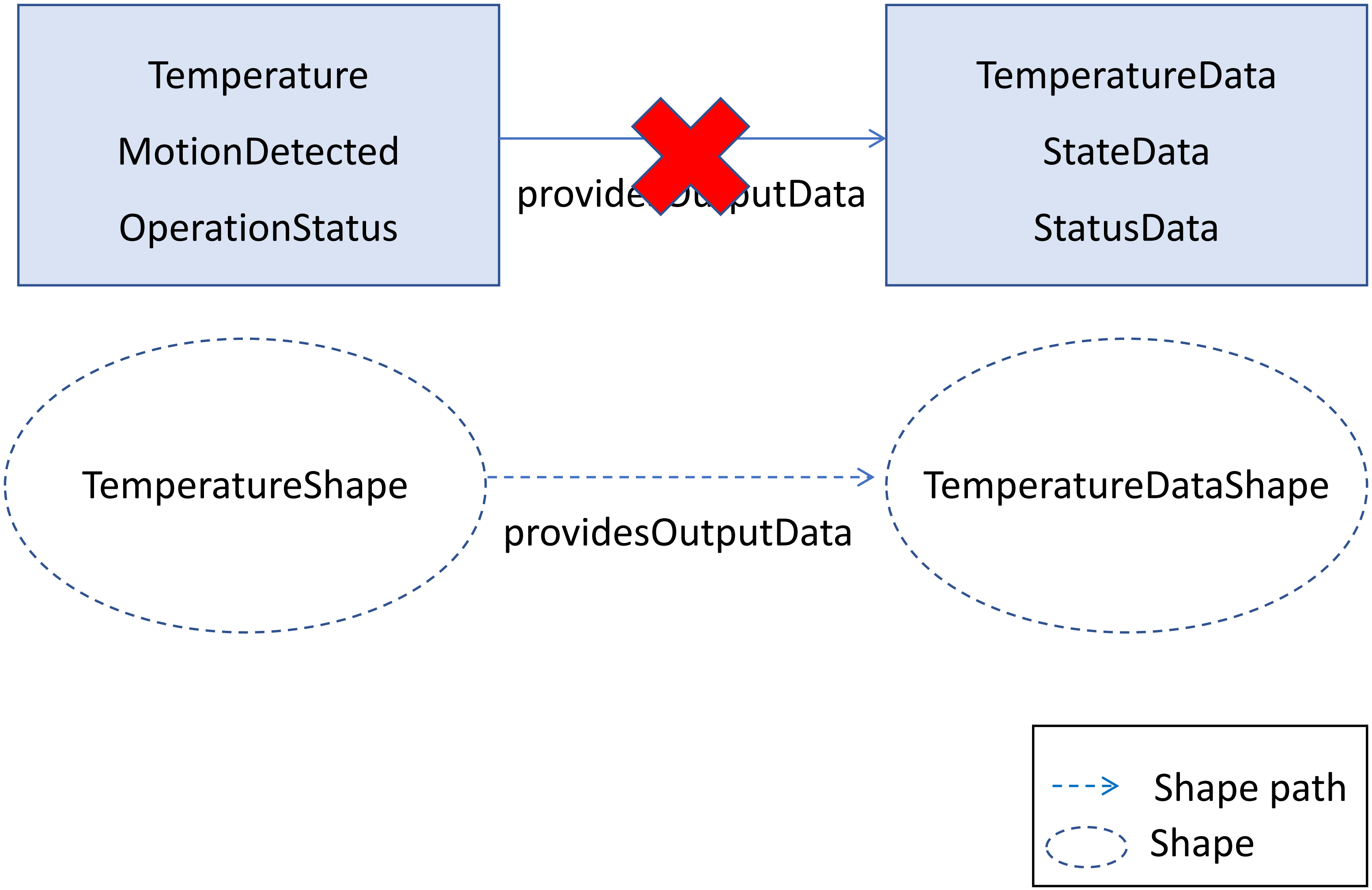
TemperatureData

StateData

StatusData

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

Proposal: Using RDF Shapes in iot.schema.org



Example: Temperature Shape

```
iotsh:TemperatureShape a sh:NodeShape ;
  sh:targetClass iot:Temperature ;
  sh:and ([
  sh:property [
    sh:path iot:providesOutputData ;
    sh:minCount 1;
    sh:maxCount 1;
     sh:node iotsh:TemperatureDataShape ;  ];  ]
 [ sh:not [
   sh:property [
   sh:path iot:acceptsInputData ;
   sh:minCount 1 ;   ]];  ]
[ sh:property [
    sh:path iot:writable ;
     sh:minCount 1;
    sh:maxCount 1;
```

Example: Temperature Data Shape

```
iotsh:TemperatureDataShape a sh:NodeShape ;
 sh:and ([
 sh:property [
   sh:path schema:propertyType ;
   sh:minCount 1; sh:maxCount 1;
    sh:datatype xsd:integer ;  ] ;  ]
[  sh:property [
   sh:path schema:minValue ;
    sh:minCount 1; sh:maxCount 1;
    sh:datatype xsd:integer;  ];  ]
[  sh:property [
   sh:path schema:maxValue ;
    sh:minCount 1; sh:maxCount 1;
    sh:datatype xsd:integer;  ];  ]
[ sh:property [
   sh:path schema:unitCode ;
   sh:minCount 1 ; sh:maxCount 1 ;
   sh:in ( iot:Celsius iot:Kelvin iot:Fahrenheit);  ]; ] ).
```

Current proposal:

https://github.com/iot-schema-collab/iotschema/blob/master/shapes/interaction-patterns%20-%20shapes.jsonld

Charter Update

- W3C Community Group Charter – WoT CG
  - Incubate semantic definitions for iot.schema.org
  - Support multiple application domains, including connected home, automotive, industrial
  - Community contribution process according to governing IPR policy of schema.org or W3C community groups
  - Use the CG mailing list for community discussions and consensus process
  - Detailed technical discussion will use github issues

Next Steps

- Get the WoT CG Charter approved
- Update the base model to include dataItem and Feature of Interest concepts
- Define dataItem constraint mechanisms
- Create prototypes for Feature of Interest definitions
- Set up process and work area for incoming definitions
- Continue to develop tools
- Develop user guidance documentation