IoT Extensions for schema.org

Community Teleconference

June 18, 2020

Agenda

- Agenda review
- Announcements and industry developments
- Status review
- Ongoing topics – new progress ?
  - Thing class
  - One Data Model integration/format conversion
  - schema.org integration patterns
- AOB

Announcements

- W3C WoT Online Virtual F2F June 15-19
  - Focus on directory integration, discovery, semantics
- OneDM preparing to emerge and "go public"
  - SDF language is stable and moving toward IETF standardization
  - ~200 models in the playground; OCF, LWM2Mp, Zigbee, BLE Mesh examples, Energy models are another focus
  - Working on public-facing content for web pages and FAQ
- New group in Building and Lighting industry to harmonize across several standards – commercial focus but may spawn a data model harmonization activity

Status update

- Implementation status

Conclusion

- Update ongoing topics
- Summary
- AOB

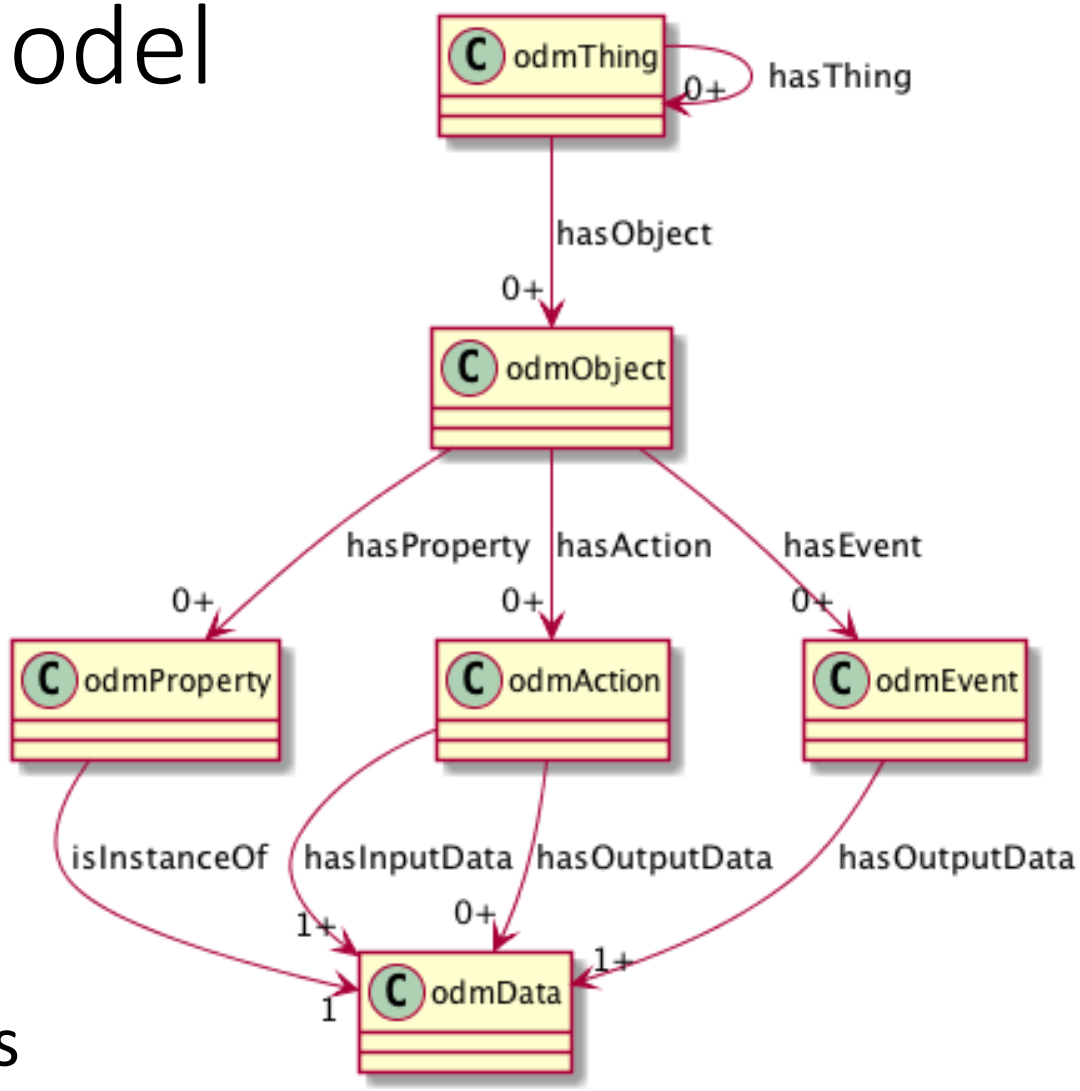
Backup

Project CHIP

- Google and Apple joined Zigbee Alliance to create a new interoperable network standard for connected homes – Project Connected Home over IP
- Deliver a standard, open source reference stack, and certification program for interoperable devices
- What it means to iotschema – unified data models for connected home devices + Event, Action, Property model + open source license
- Still need to address system level modeling with location, context , behavior + application domains

ODM Meta-Model

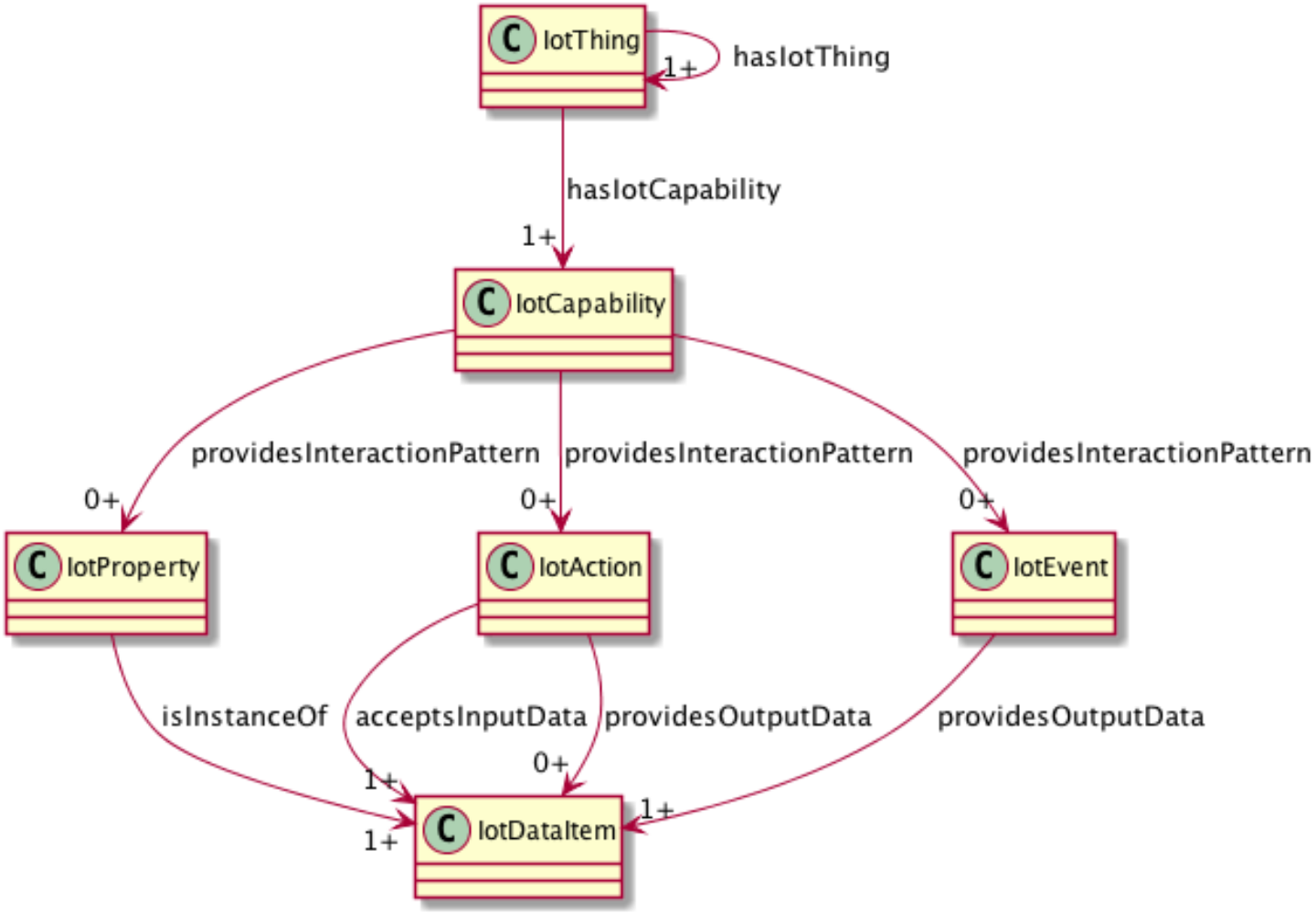
- Thing Class to compose Objects
- View (Interface) Class to virtualize affordances
- Reusable Objects
  - Property, Action, and Event Affordances
- Reusable Data Types



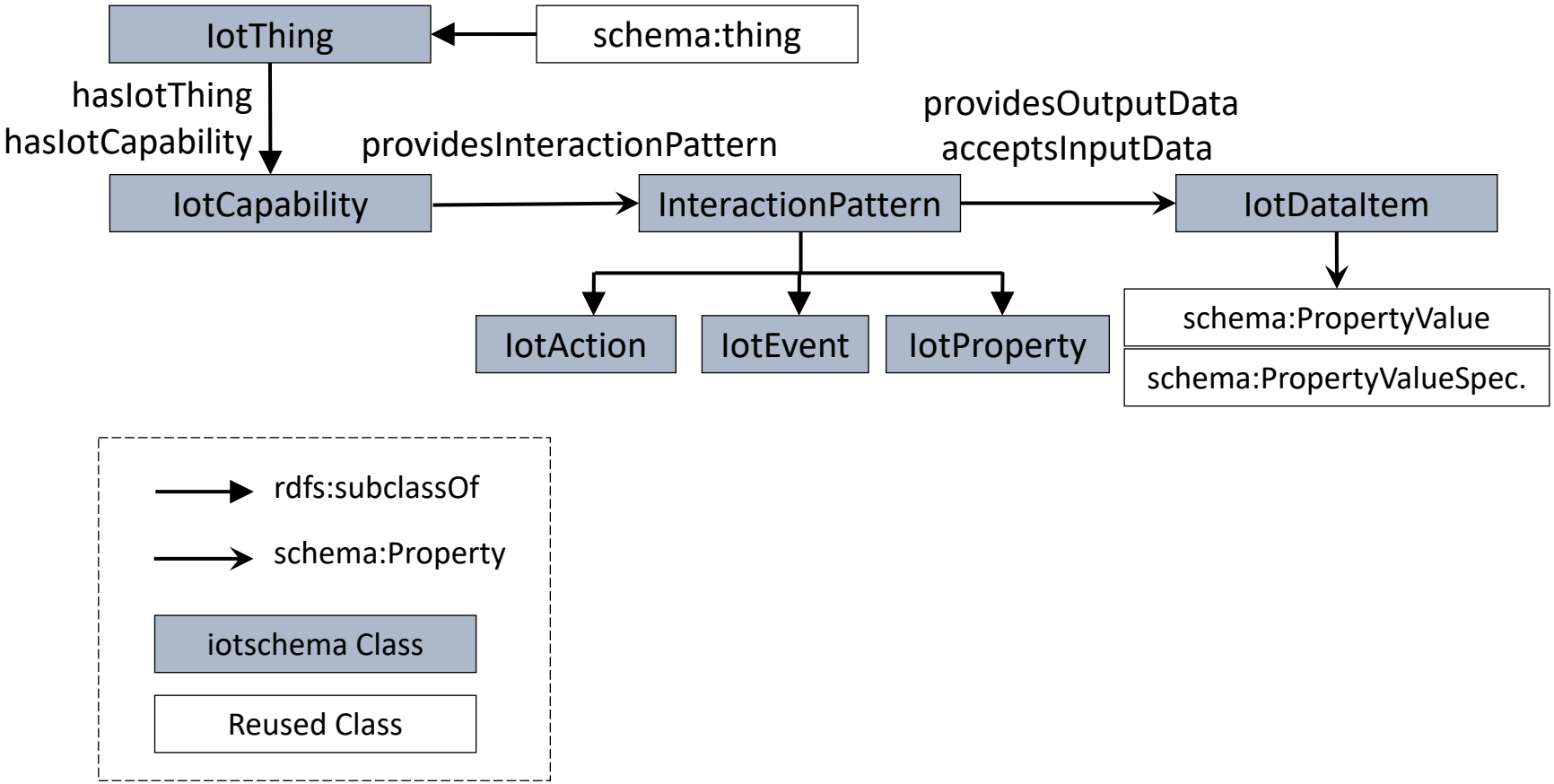
Thing Class

- Encapsulate reusable Capabilities
  - On/Off with state Property, Commands, and Events
  - Compose Air Conditioner Thing from OnOff, Mode, Speed, etc. as reusable Capabilities
- Reusable compositions of Capabilities
  - An Outlet unit for a multi-outlet strip
  - Each Outlet has OnOff, Energy Monitor, Overcurrent and Overtemperature protection Capabilities
  - Multiple Outlets are composed into an outlet strip
  - Outlet unit can be a Thing
  - Outlet Strip can also be a Thing

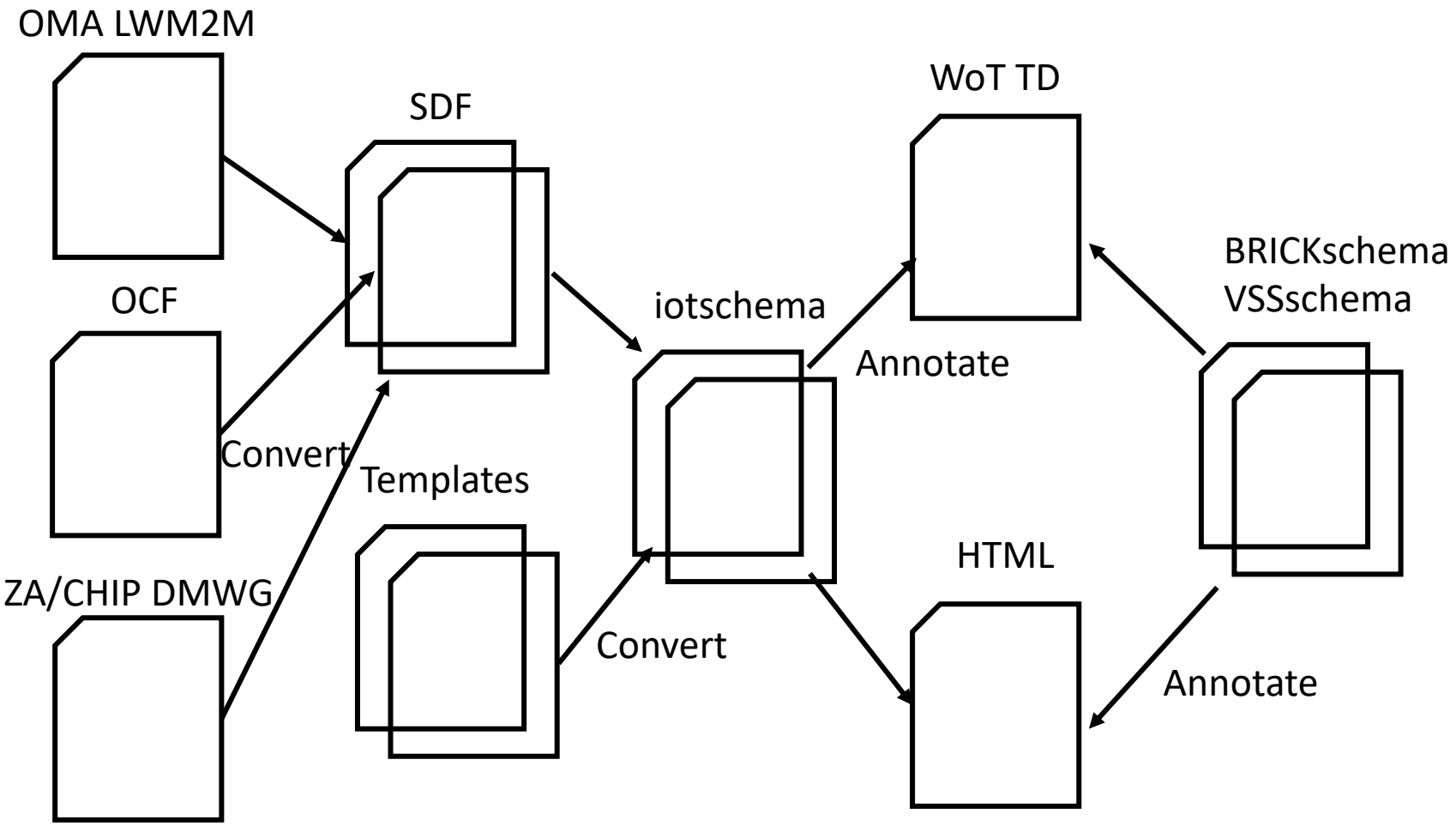
iotschema UML with iotThing class



schema.org IoT Extension Meta Model with Thing Class



High Level Integration Patterns



One Data Model integration

- Re-shape One Data Model definitions as iotschema definitions
- One Data Model uses JSON object hierarchy vs. RDF links
  - JSON pointer fragment identifiers
  - st:#/sdfObject/onoff/sdfAction/turnon
- iot:onoffCapability => iot:providesTurnonAction => iot:turnonAction
  - iot:providesTurnonAction is a sub-class property from iot:providesInteractionPattern

iotschema from OneDM

- st:#/sdfObject/onoff/sdfAction/turnon
- sdfObject is "sameAs" iotCapability
- Create type names
  - st:#/sdfObject/switch => iot:switchCapability
  - st:#/sdfObject/switch/sdfAction/turnon => iot:turnOnAction (iot:switchTurnOnAction?)
- Synthesize the schema.org style property types
  - providesInteractionPattern subtypes
  - *providesSwitchTurnOnAction*

Path Construct in RDF

- st:#/sdfObject/onoff/sdfAction/turnon
  - "@id": "iot:iotCapability/onoff"
  - "@id": "iot:iotCapability/onoff/iotAction/turnon"
- What does the property type look like?
  - iot:providesInteractionPattern
  - iot:providesTurnonAction
  - iot:providesiotCapability/onoff/iotAction/turnon

Schema.org Integration

- Class names Event, Action, Property conflict
- iotschema has diverse semantic types for objects, schema.org has diverse property types
- Property types could be synthesized from objects but…
- iotschema will potentially define hundreds of types for physical quantities (temperature, humidity, voltage, acceleration…), control affordances (open/close, brightness, color control, camera controls, operating modes…), and features of interest (rooms, machines…)

Schema.org Integration

- The WoT use case is based on annotation consisting of RDF @type statements that point to URIs of defined terms for specialized types that conform to the classes in the meta-model
- These meta-model classes would only add about 6 new property types to schema.org
  - iotCapability, iotEvent, iotAction, iotProperty, iotData, iotFeatureofInterest
  - new types like iotInterface, iotThing, etc. as needed

Schema.org Integration

- There is a potential example pattern in schema.org
  - MedicalEntity, with about 7 property types
- Likewise, an IoT Schema instance would contain some set of iotCapability, iotAction, iotProperty, iotEvent, iotFeatureOfInterest properties
- Specialization of iot types would happen at the next level in the graph – hosted in a separate namespace
  - URIs that point to accepted specialized definitions in one or more specialized namespaces
  - lighting controls, thermostats, etc. that conform to the base types but have their own properties