Web of Things

Linked Data & Semantic Processing Task Force
Osaka Face to Face, May 2017

Dave Raggett <dsr@w3.org>

Semantic Interoperability

- Semantic Interoperability is increasingly a priority to enable open markets of services
  - Ensuring communicating parties share the same meaning for the data that they exchange
- Existing IoT standards suites focus on the protocols and interaction patterns, and only informally describe the semantics in the prose text of the specifications
- Some background on semantic models
  - OneM2M ontologies
    - http://www.onem2m.org/technical/developers-corner/tools/onem2m-ontologies
  - IEEE IEEE Standard Ontologies for Robotics and Automation,
    - https://standards.ieee.org/findstds/standard/1872-2015.html
  - W3C Semantic Sensor Network
    - https://www.w3.org/TR/vocab-ssn/

Semantic Interoperability

- Interaction model for things expressed in term of the software object properties, actions and events
  - Data types and constraints, e.g. min/max
  - Units of measure, e.g. degrees Celsius
- Links to semantic models
  - Support for discovery, composition, validation, and adaptation to variations across devices
  - We now need to work on how to realise this!

Let’s get practical …

- OCF, oneM2M and ECHOnet all specify devices for smart homes, but they vary in the details of the interaction models and capabilities
- Let’s look at some specific examples and discuss ideas for how to express the corresponding semantic models

Same devices, different capabilities

- Let's compare the different interaction models for OCF and oneM2M devices
  - **OCF devices**: air conditioner, air purifier, window blind, camera, dishwasher, door open status, dryer, fan, garage door, on/off light, oven, printer, printer multi-function, receiver, refrigerator
  - **oneM2M devices**: air conditioner, clothes washer, electric vehicle charger, smart light, electrical generator, oven, refrigerator, robot cleaner, electric meter, storage battery, television, thermostat, water heater.
- The definition of an air conditioner is very different between the two platforms. OCF just defines an on/off switch and a temperature range. oneM2M, however, also defines the step interval for temperature, a turbo mode, a run mode with a set of states, a timer, and a wind speed setting expressed via an enumeration.

More details

- Full details of OIC 1.1 and oneM2M Home Appliances:
  - https://www.w3.org/WoT/demos/td2ttl/oic.html
  - https://www.w3.org/WoT/demos/td2ttl/m2m.html
- These include simple JSON representations of the interaction models that have been reverse engineered from the OCF and oneM2M specs
- The demos include scripts that map the JSON to RDF as either Turtle, JSON-LD or a graphical representation
- The demos don’t yet include the semantic models …

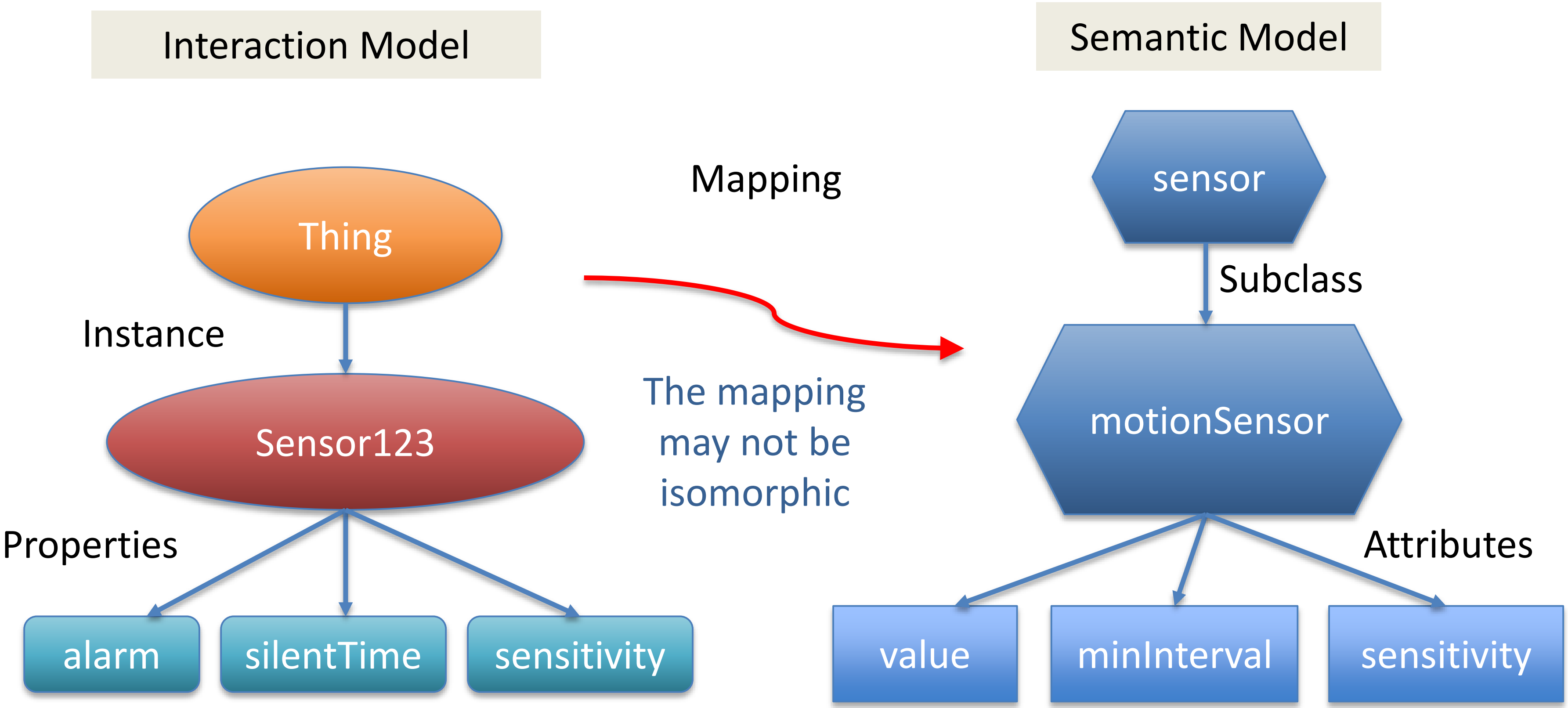
Example: Motion Sensor

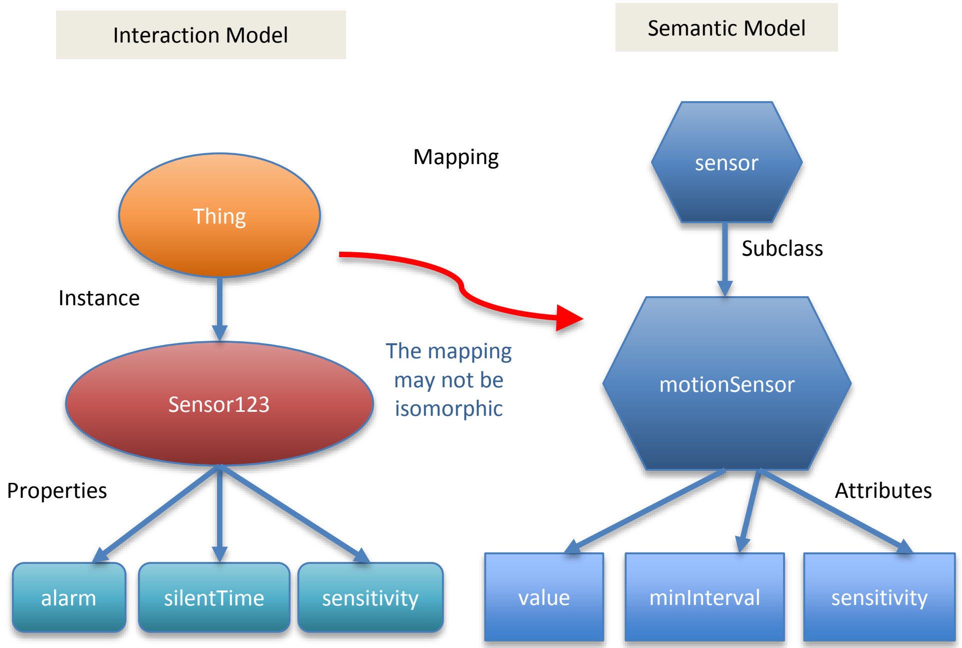
- OCF: a read-only Boolean property
  - This resource describes whether motion has been sensed or not. The value is a Boolean. A value of 'true' means that motion has been sensed. A value of 'false' means that motion not been sensed.
- oneM2M: further properties
  - Wait time in case of continuous motion
  - Integer value for detection accuracy

Semantic Model

- Thing is an instance of a motion sensor class
  - Instances of this class may have a wait time
  - Instance of this class may have an adjustable sensitivity
- Thing descriptions from different vendors may use different names for the properties, actions and events, and moreover, there will be variations in the capabilities available
  - Vendors want to differentiate their products from their rivals
  - OCF, oneM2M, ECHOnet, etc. all define smart home devices differently
  - The Web of things needs to support such variations
- How to cope with different property names for essentially the same concept?
  - One idea is to assert that the property is an instance of the motion sensor class
    - The wait time and sensitivity would then need to be exposed in the interaction model as read/write metadata for that property
  - Another idea is to assert the **semantic role** of each property
    - The thing is declared as an instance of the motion sensor class
    - The oneM2M alarm, silentTime and sensitivity properties are declared with their respective semantic roles

Defining a Mapping




Interaction Model
Semantic Model
Mapping
Thing
Instance
Sensor123
Properties
alarm
silentTime
sensitivity
The mapping may not be isomorphic
sensor
Subclass
motionSensor
Attributes
value
minInterval
sensitivity

Some Comments

- If the interaction model is represented in JSON we can use @context to map strings to URIs
- This allows us to translate JSON to RDF
- But interaction model is not same as semantic model, so conflating the two is likely to cause problems
  - e.g. when risk of invalid reasoning when mapping several different interaction models to the same semantic model
  - Thus need for explicit mapping, hence "semantic role"

Can Defaults Help?

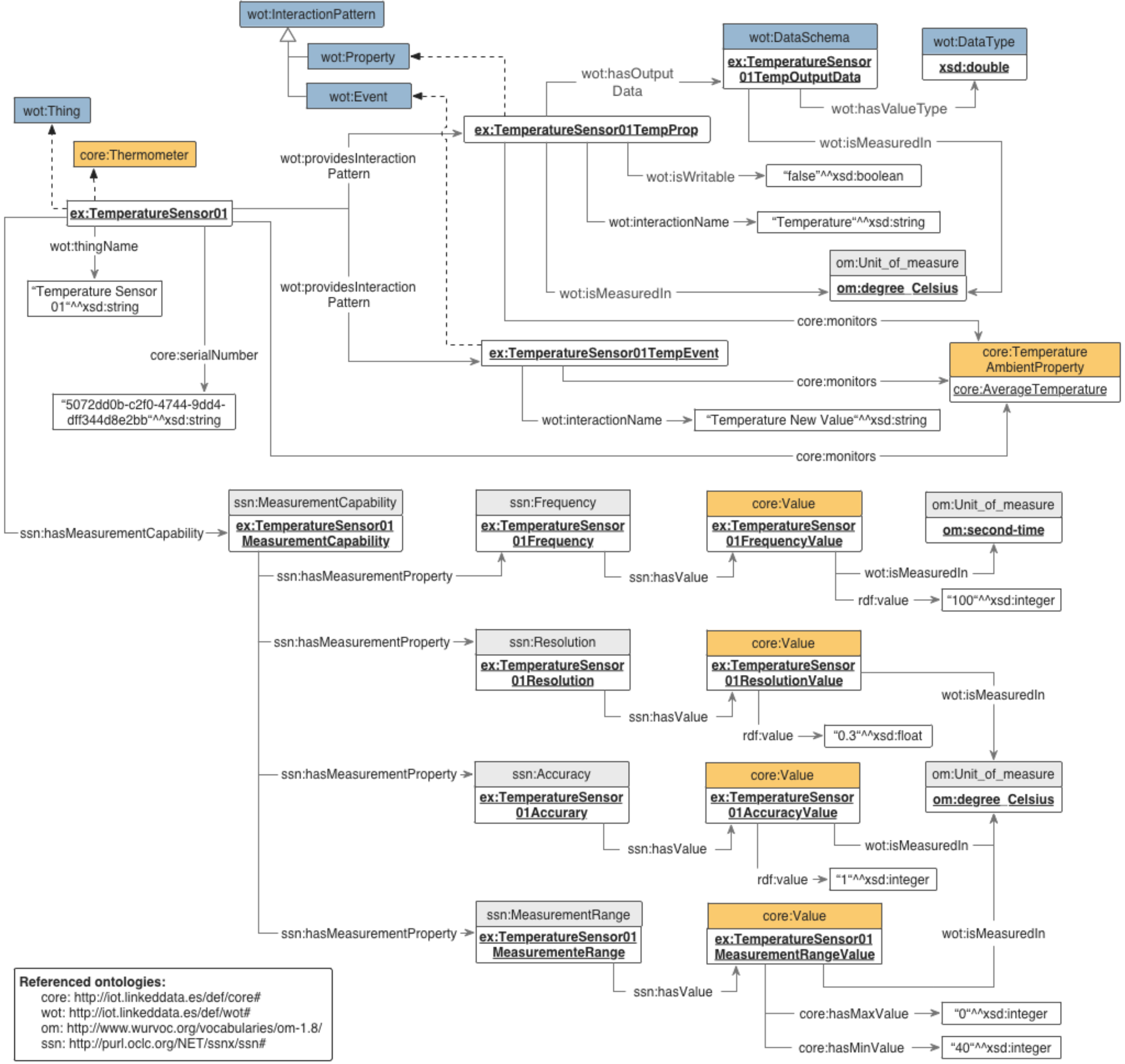
- Opportunity to simplify thing descriptions via the use of defaults in the semantic models
  - Units of measure
    - Where the same units are used in majority of instances of a particular semantic class
  - “standard” property names
    - Using standard name for a property avoids need to explicitly declare its “role” in interaction model
- Inheritance from super classes
  - e.g. declarations on the generic “sensor” class

Linked Data Technologies

- Keeping it simple with RDF Schema together with additional predicates
- OWL ontologies
  - Increased sophistication and complexity
- Validation with Linked Data Shape rules
  - Potential for simple graphical rules
    - SHACL, ShEx, SHRL as different flavours of shape rule languages
- Other tools
  - Semantic Data annotating tools
  - Storage and query engines
  - Reasoners
  - …
- But we shouldn’t be scared of introducing new approaches where these make obvious sense

Semantic Models

- We need to win over people who are suspicious of semantic technologies!
  - Widespread use in research projects
  - But comparatively little commercial adoption
- We need to show that semantic interoperability is easy and solves commercially important challenges


wot:InteractionPattern
wot:Property
wot:Event
wot:Thing
core:Thermometer
ex:TemperatureSensor01
wot:thingName
"Temperature Sensor 01"^^xsd:string
core:serialNumber
"5072dd0b-c2f0-4744-9dd4-dff344d8e2bb"^^xsd:string
wot:providesInteraction Pattern
wot:providesInteraction Pattern
wot:hasOutput Data
wot:DataSchema
ex:TemperatureSensor 01TempOutputData
wot:DataType
xsd:double
wot:hasValueType
wot:isMeasuredIn
ex:TemperatureSensor01TempProp
wot:isWritable
"false"^^xsd:boolean
wot:interactionName
"Temperature"^^xsd:string
om:Unit_of_measure
om:degree_Celsius
wot:isMeasuredIn
core:monitors
core:Temperature AmbientProperty
core:AverageTemperature
ex:TemperatureSensor01TempEvent
core:monitors
wot:interactionName
"Temperature New Value"^^xsd:string
core:monitors
ssn:hasMeasurementCapability
ssn:MeasurementCapability
ex:TemperatureSensor01 MeasurementCapability
ssn:hasMeasurementProperty
ssn:Frequency
ex:TemperatureSensor 01Frequency
ssn:hasValue
core:Value
ex:TemperatureSensor 01FrequencyValue
om:Unit_of_measure
om:second-time
wot:isMeasuredIn
rdf:value
"100"^^xsd:integer
ssn:hasMeasurementProperty
ssn:Resolution
ex:TemperatureSensor 01Resolution
ssn:hasValue
core:Value
ex:TemperatureSensor 01ResolutionValue
wot:isMeasuredIn
rdf:value
"0.3"^^xsd:float
ssn:hasMeasurementProperty
ssn:Accuracy
ex:TemperatureSensor 01Accurary
ssn:hasValue
core:Value
ex:TemperatureSensor 01AccuracyValue
om:Unit_of_measure
om:degree_Celsius
wot:isMeasuredIn
rdf:value
"1"^^xsd:integer
ssn:hasMeasurementProperty
ssn:MeasurementRange
ex:TemperatureSensor01 MeasurementeRange
ssn:hasValue
core:Value
ex:TemperatureSensor01 MeasurementRangeValue
wot:isMeasuredIn
core:hasMaxValue
"0"^^xsd:integer
core:hasMinValue
"40"^^xsd:integer
Referenced ontologies:
core: http://iot.linkeddata.es/def/core#
wot: http://iot.linkeddata.es/def/wot#
om: http://www.wurvoc.org/vocabularies/om-1.8/
ssn: http://purl.oclc.org/NET/ssnx/ssn#

Simple?

Roadmap?

- Could we define a roadmap for demonstrating benefits of semantic technologies?
  - NodeJS implementation on Web of Things with access to simulations of OCF, oneM2M and ECHOnet devices?
  - Smart services that adapt to variations in the interaction models based upon inspecting their semantic models
  - Validation of interaction models are consistent with their linked semantic models
  - Virtual things as dynamic compositions of other things based upon a registry of services
- Practical way to explore different approaches to representing semantic models

Requirements for Semantic Models

- The means to declare semantic classes
  - Taxonomies of semantic classes
  - Constraints on interaction models
    - Properties, actions, events, metadata
    - Whether optional or required
  - Use of roles for identifying properties, actions and events independent of the name used in specific interaction models
  - Default names for properties, actions and events
  - Defaults for units of measure
- Don’t forget the overriding need to keep it simple!
  - Even if this implies the need for new standards

Challenge!

- Discuss how to address the requirements for semantic models of things for the oneM2M motion sensor
  - How to represent the semantic model?
    - e.g. the need for a property with a given role, and the means to express defaults
  - Whether current standards support the kinds of reasoning required?

Different Communities
Different Semantic Models

- When different communities work on defining semantic models we can expect differences
- This is already the case in the research community, and can be expected to be the case for Standards Development Organisations
  - Evidence from comparing OCF, oneM2M and ECHOnet
- Tools relating to mappings between models
- What social and technical factors can encourage reuse and convergence?

Some ideas for discussion

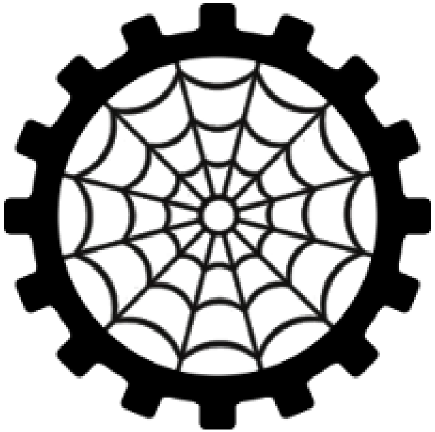
```
:motionSensor rdfs:subClassOf :sensor .

:motionSensor td:hasInteractIonModel _:23 .

_:23 td:hasProperty _:31 , _:32 , _:33 .

_:31 td:role “value” ;
        rdfs:comment “true if motion has been detected” ;
        td:type td:boolean ;
        td:writeable false .

_:32 td:role “minInterval” ;
        rdfs:comment “minimum interval between alarms” ;
        td:type td:integer ;
        td:optional true .

_:32 td:role “sensitivity” ;
        rdfs:comment “detector sensitivity” ;
        td:type td:integer ;
        td:optional true .
```

- Importance of graphical representation
  - As a tree of nodes and attributes
- motionSensor is a sub-class of “sensor”
- motionSensor has an interaction model
- Each property is identified by a role
  - A string literal
- Properties may be required or optional
- Properties may have metadata
- Properties may have default names
- This example needs extending to declare the units for minInterval and sensitivity
- The semantic model can be validated against a specific interaction model
  - Using a simple script and Linked Data library
- Questions
  - Would OWL be simpler or more complex?
  - Which requirements would OWL leave unfulfilled?

Units of Measure

- JSON context maps names to RDF concepts
  - Local context to disambiguate short names
    - mA for milliamperes
    - Grouping by domain and system (e.g. SI vs Imperial)
- RDF concept for combination of unit of measure and scale factor, e.g. milliamperes (amperes x 1000)
- Concept acts as link to further triples that identify
  - the base unit (amperes)
  - the scale factor (1000)
  - the property being measured (electrical current)
  - Conversion formulae between different units
- Questions, e.g. relationship to QUDT, and whether the WoT IG should survey the needs for common domains, e.g. smart homes and make some recommendations for standardisation

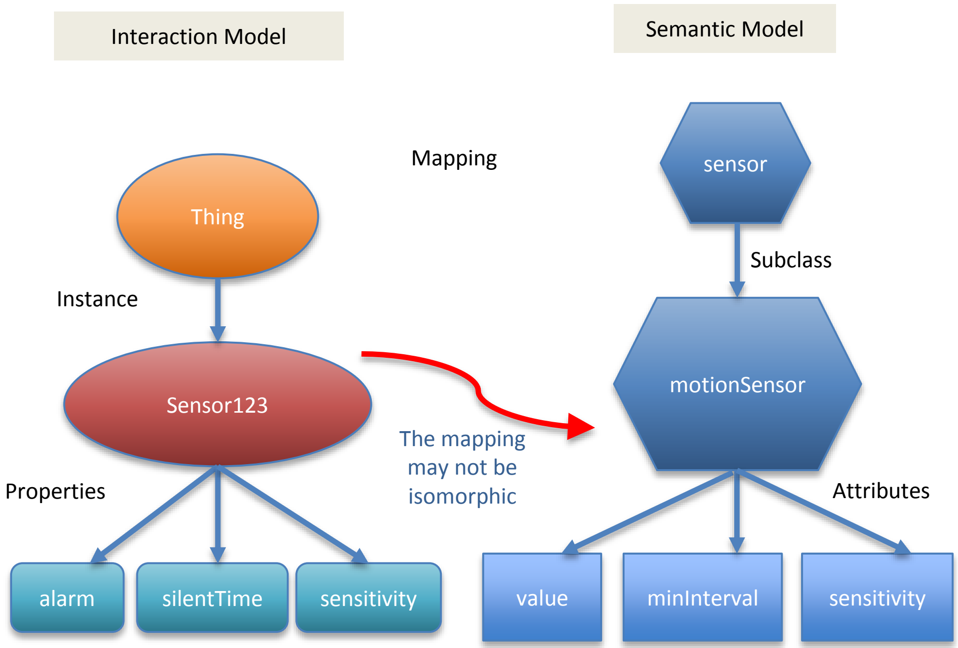
Epilogue – Cognitive Web



- Semantic models for the Web of Things are the starting point for the Cognitive Web
- Extension to Linked Data to support reasoning more like we humans do
  - Synthesis of AI and Cognitive Science (ACT-R)
  - Based upon statistics of prior experience
  - Link strengths and exponentially decaying activation levels
  - What-when, what-if and semantic knowledge
  - Cognitive rule language for procedural knowledge
  - Reasoning at multiple levels (Minsky)
  - Trained and assessed using lessons
  - Self aware cognitive agents

Summary of TF-LD session

- We focused on relationship between interaction models and semantic models, and a scalable approach based upon commercial reality
  - SDO's won't fully converge
  - Vendors need to differentiate
  - Thus need for bridging ontologies
  - W3C to define framework for linking to semantic models, and building a shared mindset across SDOs and IoT communities
  - Looking forward to exploring greater use of semantic technologies in future plugfests
- We had a valuable discussion and the task force chairs will now work on a plan for a roadmap with clearly defined short term goals in the run up to the next Face to Face
- This includes a study of existing work, draft ontologies* for OCF, ECHOnet & OPC, tooling, and opportunities for W3C to define APIs for accessing semantic context



* oneM2M has already defined their ontologies

Thanks