Schema.org
Extensions for IoT
(iotschema)

Darko Anicic, Michael Koster
Data Models and Semantic Interoperability
IoT Week, Aarhus, Denmark - 20.06.2019
Data models for IoT

- Web of Things
  Thing Description
  W3C Candidate Recommendation

- Few other IoT ecosystems revolve around similar thing data models.

The AWS IoT Things Graph Data Model (TDM)

The Alibaba Thing Specification Language (TSL)

Web Thing Description
Common semantic layer

iotschema.org
Semantic interoperability for connected things

W3C
Web of Things Thing Description

aws
The AWS IoT Things Graph Data Model (TDM)

Alibaba Cloud
The Alibaba Thing Specification Language (TSL)

WebThings
Mozilla
Web Thing Description
What is iotschema?

- An open, publicly available, repository of semantic definitions for connected things
- An extension of schema.org to enable descriptions of things in the physical world and their data
- A common set of tools and patterns, and a community process for contribution and publication of standardized definitions
- A way for domain experts to easily create semantic definitions that are relevant to their application domain
What is iotschema (2)?

- A layer to bridge between device ecosystems and Semantic Web technology
- Property and relation types to enable reuse of existing ontologies and definitions
  - SSN, SOSA, SAREF, QUDT
  - Property types for e.g. Feature of Interest
- Annotation vocabulary for WoT Thing Description
  - Common definitions for application-specific Events, Actions, and Properties
Who is iotschema for?

• **IoT platform providers** will use iotschema to make it easy for third party applications to use the platform

• **Device vendors** and SDOs will use iotschema to publish protocol-neutral definitions of their devices to enable web scale adoption

• **Domain experts** will use iotschema to create domain-specific languages for connected things and their applications

• **Application providers** will use iotschema to make their applications portable across platforms
iotschema: Semantic categories

- **iotschema** semantic definitions consist of three categories, or classes, that describe a measurement or actuation, of some physical property or item
  - A **Capability** describes the smallest practical compose-able unit of functionality (measurement an/or actuation), e.g. the temperature of something, or the brightness of a light bulb. A Capability has some related Interactions.
  - An **Interaction (Event, Action, or Property)** describes an affordance to the capability, which may be to read or write a value, or perform a complex action.
  - **Data Item** descriptions contain data types, units, minimum and maximum values, and other information about the data model, for example a shape or schema.
iotschema: Capability pattern

- `schema:thing`
- `providesInteractionPattern`
- `providesOutputData`
- `acceptsInputData`

- `Capability`
- `InteractionPattern`
- `DataItem`
- `Action`
- `Event`
- `Property`
- `schema:PropertyValue`
- `schema:PropertyValueSpec.`

- `rdfs:subclassOf`
- `schema:Property`

- `iotschema Class`
- `Reused Class`
iotschema: Example

[Diagram showing the relationship between IoT schema components]
iotschema: Feature Of Interest pattern
iotschema: Conceptual integration with other ontologies

• Feature of Interest concepts and property types to describe location, equipment, or other classifiers
• For example, BrickSchema definitions from Haystack
• Quantity and Units constraints can use QUDT concepts and appropriate identifiers
• SSN, SOSA, SAREF concepts can extend a definition
• W3C Linked Data Working Group
Connect things to the real world

"Lock Security Doors and Check"

Door Lock

Door

DoorLock Capability
- ActuateLock Interaction
- ActuateUnlock Interaction
- GetState Interaction -> LockState Data

isAssociatedWith

Is A Front Door
- Opens To Outside
- Is A Security Door

iot.schema.org

other ontologies
How to contribute to iotschema?

https://github.com/iot-schema-collab/iotschema/tree/master/incoming
Status

• Monthly Teleconferences since mid-2017
• Examples of Definitions in a Github repository
• FoI annotation examples are also in the repo
• Prototypes tested at W3C Web of Things Plugfests and WISHI/IETF Hackathons from mid 2017
• Contributors are ready to begin submitting definitions
• Next steps are to build out tools and processes
• W3C Community Group
Current members
W3C CG: Schema.org Extensions for IoT

Fujitsu  Siemens  Berkeley Lab

Google  Samsung SmartThings

Schneider Electric  INRIA  Ericsson

Universidad Politécnica de Madrid
iotschema: Resources

• W3C Community Group:
The Schema Extensions For IoT
  • https://www.w3.org/community/iotschema/

• GitHub repository:
  • https://github.com/iot-schema-collab/iotschema
  Teleconferences:
  • https://github.com/iot-schema-collab/teleconferences
  Contributions:
  • https://github.com/iot-schema-collab/iotschema
  Charter:
  • https://github.com/iot-schema-collab/ws-charter

• Web site:
  Current location
  • http://iotschema.org/docs/full.html
  Future location
  • http://iot.schema.org

• Tools:
  iotschema for Node-RED
  • https://github.com/iot-schema-collab/iotschema-node-red
iotschema for Node-RED

Recipe-based applications
iotschema for Node-RED
Recipe-based applications

• iotschema embedded in Node-RED tool
  • Enables an easy configuration of things using iotschema definitions

• Easies the use of semantics for IoT developers
  • No need for a developer to know RDF(S), JSON-LD, RDF Shapes ...

• Simplify creation of applications with W3C WoT
  • Avoids translations of serializations formats, data types, units ...

• Demonstrates semantic discovery and processing
  • Integrates WoT Thing Directory
Example: Controlling Carbon Dioxide
Node-RED Application with W3C WoT Things
Example: Controlling Carbon Dioxide
Node-RED Application with W3C WoT Things
iotschema: Semantic Mark-Up for W3C WoT Thing Description

Node-RED diagram with highlighted properties of a KETI-CO2-Sensor node: Name, Interaction Pattern Type, Capability, Feature Of Interest Type, Feature Of Interest, Property Type, Unit Code, and Observable.
Semantic Recipe
Reusable Flow Template

MatchMaker:
• finds Things that can implement the Recipe
• based on TD with iot.schema.org mark-ups
Semantic Discovery of Recipe Ingredients

Ingredients are Node-RED Nodes
Example: Controlling Carbon Dioxide
Node-RED Application with W3C WoT Things

Wire discovered Node-RED nodes
Back-Up
Narrow Waist in System Design

Many Applications. Local and Remote

Common Infrastructure (Protocols, Formats, and Meta Models)

Many Devices, Different Ecosystems
Diverse Devices and Applications, Common Protocols and Semantics

- Diverse Applications
- Web of Things: Narrow Waist of Semantics
- Diverse Devices
- Internet of Things: Narrow Waist of Protocols

Applications
- Semantic Vocab
- Thing Description
- Protocol Binding
- Device Ecosystems
- Protocols, Formats
- Transport
- Networks

Interoperable Applications
- iotschema
- W3C Thing Description
- OCF, OMA, Zigbee, Fairhair
- IETF CoAP, CBOR, Link-Format
- UDP/TCP
- WiFi, IPV6, Bluetooth
How is iotschema used?

• Annotation of Thing Descriptions (W3C Web of Things)

• Thing Descriptions have Action, Event, and Property Interaction definitions that can be annotated with iotschema Interaction class terms

• Thing Descriptions have DataSchema elements that can be annotated with iotschema Data Item class terms and constraints, such as data type, units

• Thing Description enables applications to interact with connected things independent of protocol and SDO profile
Definitions

• Semantic definitions that follow the design patterns and interaction affordances of connected thongs
• Interoperable due to a set of static and dynamic semantic constraints
• Define a "Capability" that represents – typically – the smallest practical compose-able unit of functionality
• For example, a temperature sensor, or a door lock
Capability Model

iot.schema.org

Diagram:
- Capability
  - providesInteractionPattern
  - rdfs:subClassOf
- Interaction Pattern
  - Event
  - Action
  - rdf: Property
  - acceptsInputData
  - providesOutputData
- Data Schema
  - alignment
Feature of Interest Integration

• **Features Of Interest** (FoI) describe the real-world targets of sensing and actuation

• Definitions may be developed in iotschema, or more likely will come from domain experts
  - GENIVI/VSS is a Specification for Automotive Features of Interest, called Branches, and actuation/measurement points, called Attributes and Signals
  - BrickSchema is an adaptation of Haystack that defines Features of Interest of buildings and actuation or measurement points

• **iotschema** defines relationships between Capabilities and Features of Interest to describe **connected physical systems**