Semantic Annotation and Handling of (meta) Data in IoT Information Models

W3C WoT Open Day Presentation
Mar 26, 2018

Milan Milenkovic, IoTsense LLC and Intel Corporation
milan@iotsense.com, milanx.milenkovic@intel.com

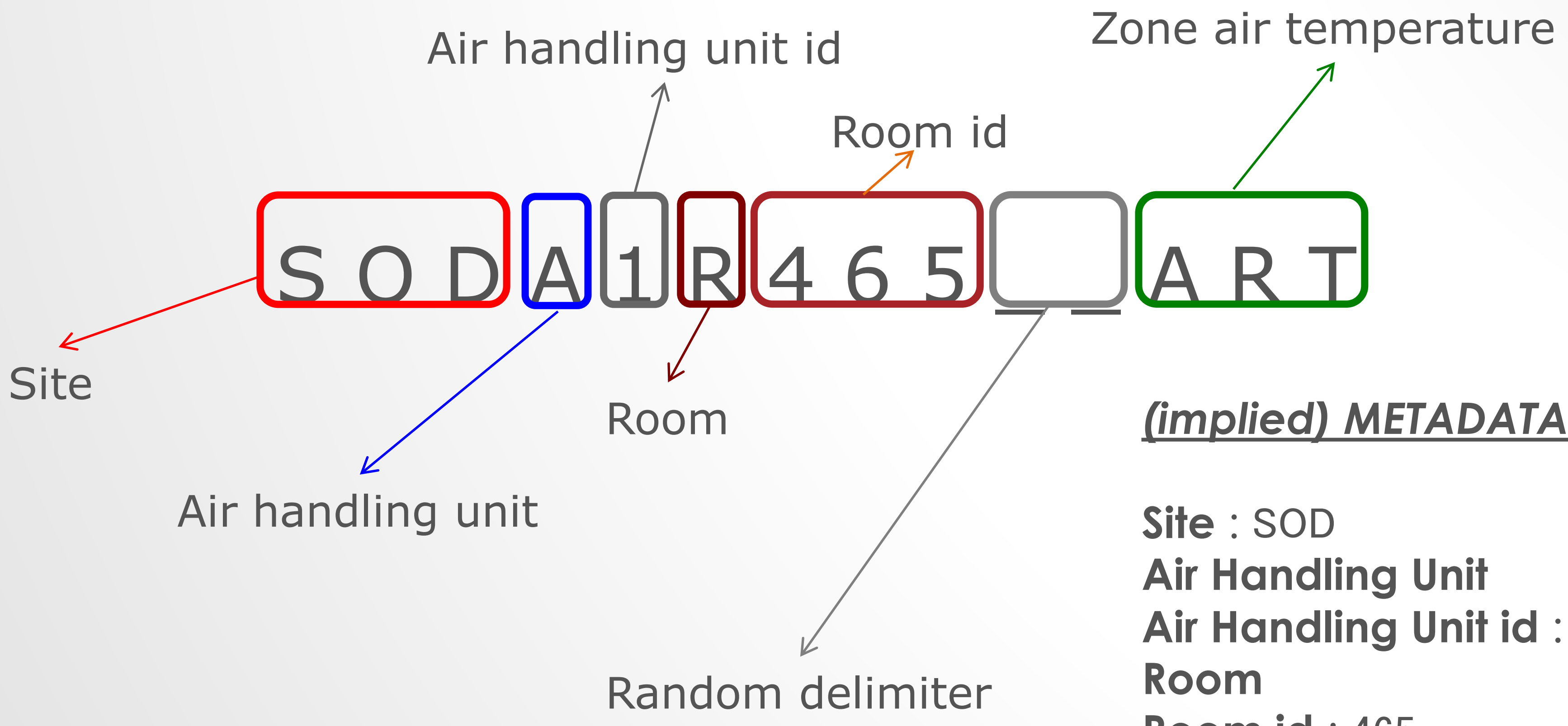
Key Messages & Outline

- Semantics/metadata is a key requirement for IoT growth
  - For (digital) Internet transformation of control systems
  - Development of third-party portable apps, big-data, AI…
- IoT standards approaches to metadata are inadequate
  - Prescriptive, limited set of attributes
  - Little or no provision for (really important) metadata later in the lifecycle
    - Installation, commissioning
- This talk (outline)
  - What & why presentation – not how
  - Industrial-grade examples of need for, value of systemic metadata
  - Some learnings and observations
  - Extras, opt. discussion – semantic schema evaluation, ex. portable app

(Building) Industry Common Situation

- Commercial buildings, managed by BMS
  - Complex HVAC machinery – chillers, heaters, pumps, valves, zones, thermostats
  - Other systems – lighting, elevators, fire, security, occupancy
  - Large number of sensors and control points, "smart" 10s thousands
- Current practice: no (reusable) semantic annotation
  - Point names assigned arbitrarily by installers/integrators
  - BMS control sequences custom tailored using those names
    - Completely obscures structural relationships, layout of equipment
    - Each building looks different, even those using the same BMS

An actual sensor in a commercial building, annotation as installed



(implied) METADATA:

Site : SOD
Air Handling Unit
Air Handling Unit id : 1
Room
Room id : 465
Zone air temp sensor : ART

Ad hoc, bespoke – no app portability across buildings, BMS…

Problems and IoT Solution?

- Problems (pre-Internet…)
  - Expensive, brittle, obscure, error prone, not scalable, prohibitive changes
  - Cannot have portable apps and services, e.g. AI, analytics across buildings, BMS
- Need/want
  - Enable portable applications, and services
  - Enable attribute-based searches
  - (cross-domain) big-data aggregations: large scale trends and patterns
    - E.g. energy efficiency of buildings of similar size and construction in similar climates, techniques and BKMs used to achieve them
- Use semantic annotation (metadata)
  - Indicate and refine functions
  - Should provide the basis for expressing equipment structural relationships
  - e.g. for apps to map/model building, production line, factory floor – like OPC UA

Sensor data and meta-data use

- Sensor “`zn3-wwf14`”” “`77.6`” ??
- Services, analytics, benefit from additional metadata info
  - Is a zone temperature
  - Is supplied by VAV box
  - Is served by AHU-1
  - Is operated on occupancy schedule #1 (7:30 am – 6:30 PM)
  - Has an occupied setpoint of 74 F
- So app can deduce anomaly, activate VAV and AHU-1 to cool until associated temp. sensor shows compliance (zn3---)
  - Also detect rouge zones (heating and cooling simultaneously on), …

Metadata example (in Haystack notation)

```
                                  //used to denote comments, not official syntax
"id": "150a3c6e-bef0ee0e",        // (G)UID
“dis”: “zn3-wwf14”                //string, for UI display
 "sensor": "m:",                  // marker is Haystack notation for metadata
 "temp": "m:",                    // meta, measures temperature
  “air”: “m:”,                    //   of air
 "curVal": "n:77.60",             // current value
  "unit": "F",                    // measurement unit, F
 “zone”: “m:”,                    //is in a zone (same as AHU-1 in this ex.)
 “floor”: “n:4”,
 “scheduleRef”: occSchedule1,     //links, references
 “equipRef”: “@AHU-1”             //
```

Some Observations: Metadata

- Metadata provide
  - Context in which things operate and how they relate to each other
- (rich) Metadata in IoT systems enable:
  - Flexibility to add new sensors, new functions post-installation
    - e.g. increase coverage in key areas, add predictive maintenance;
  - Facilitate attribute-based search
  - Customer choice of service providers, apps – just like the web
- A lot of useful metadata comes into existence after installation
  - Location, connections, structural relationships

Some Learnings and Observations

- IoT systems need to allow expressing of metadata in all stages of system lifecycle – not just design-time info model…
- Metadata in IoT: many variations and combinations, some rarely changing = require special treatment?
  - Opt1: allow variable metadata key, value pairs in info models
    - vs. fixed object structuring with metadata as prescribed properties
  - Opt2: separate APIs/queries to fetch static metadata as an overlay?
  - Opt3: xxx?
- Descriptive, not prescriptive
  - Does not mandate which meta/tags to use with which entity BUT
  - defines how to name and structure tags when used

Key Messages & Outline

- Semantics/metadata is a key requirement for IoT growth
  - For (digital) Internet transformation of control systems
  - Development of third-party portable apps, big-data, AI…
- IoT standards approaches to metadata are inadequate
  - Prescriptive, limited set of attributes
  - Little or no provision for (really important) metadata later in the life-cycle
    - Installation, commissioning
- Advocate
  - Metadata as first-order IoT citizen (attributes)
  - Flexible, add as many as needed, anticipate changes…
  - Descriptive, not prescriptive
  - Account for all stages in lifecycle – installation, commissioning

ANALYZING METADATA SCHEMAS

An interesting approach: Analyzing Metadata Schemas – driven by application requirements

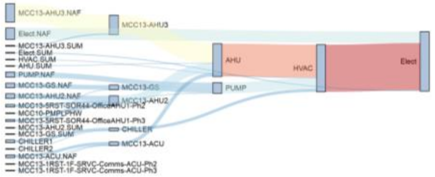
Short Paper: Analyzing Metadata Schemas for Buildings: The Good, the Bad, and the Ugly, Buildsys 2015

Authors: Arka Bhattacharya University of California, Berkeley, Berkeley, CA, USA, Joern Ploennigs IBM Research, Dublin, Ireland, David Culler, University of California, Berkeley, Berkeley, CA, USA

Applications

89 published smart-building applications were analyzed to identify the essential information required. The applications were classified in:

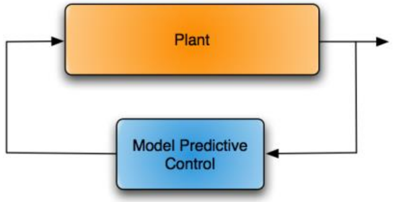
Energy Apportionment



Web Displays

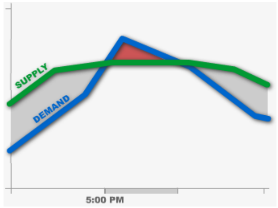
Occupancy Modelling

Model-Predictive Control



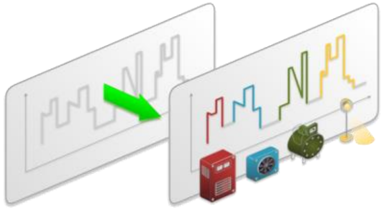
Participatory Feedback

Demand Response

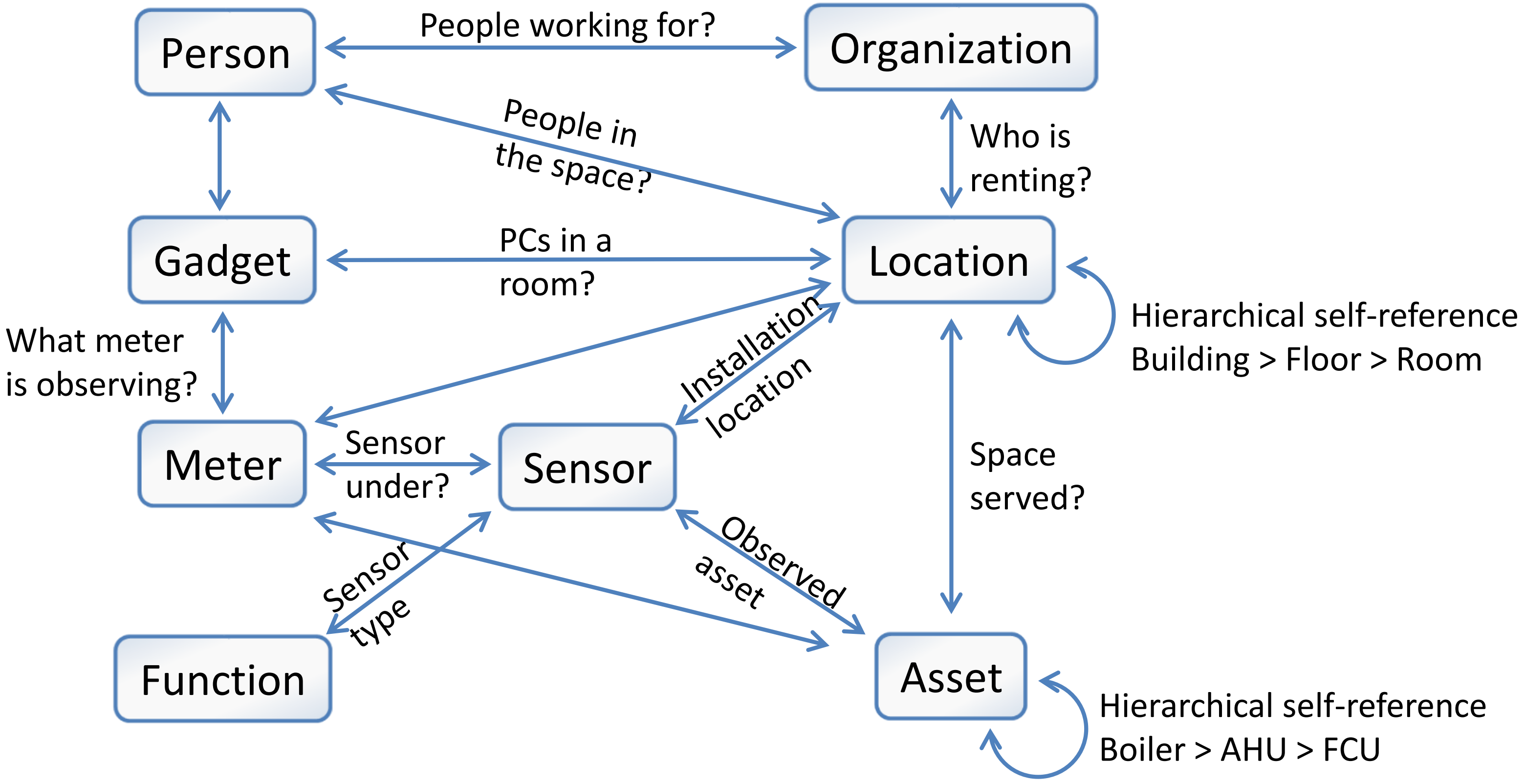


Fault Detection and Diagnosis

Non-Intrusive Load Monitoring



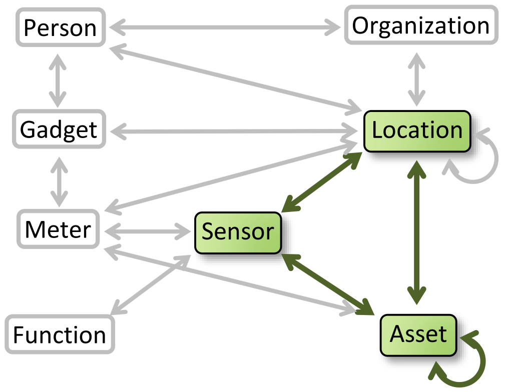
Relationships Required By Applications



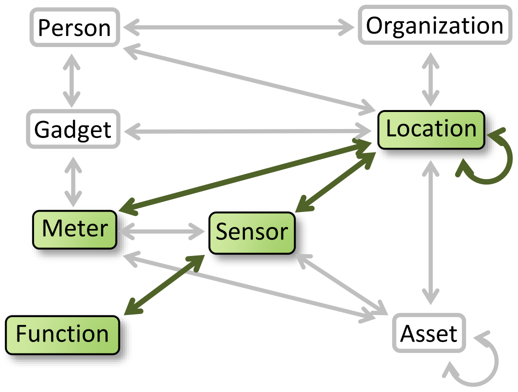
Applications Information Requirements

89 published smart-building applications were analyzed to identify the essential information required. The application list is public available.

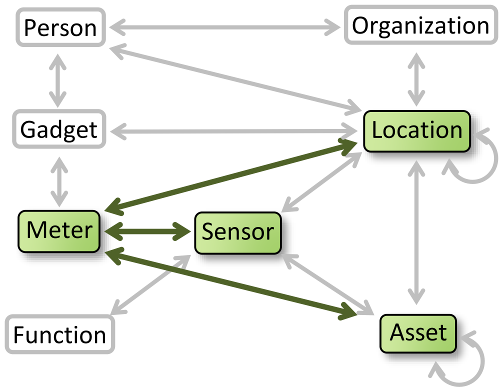
Feedback / MPC / FDD



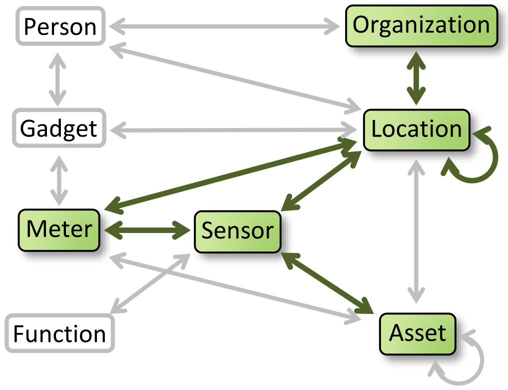
Web Display



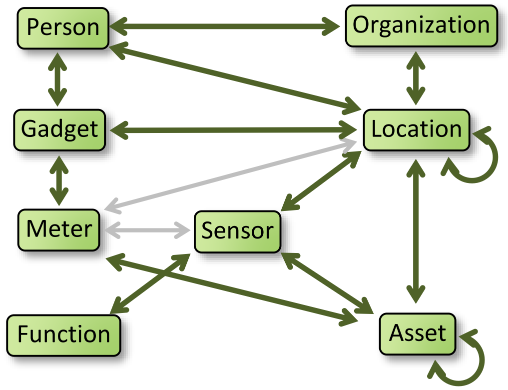
NILM / Demand Response



Energy Apportionment

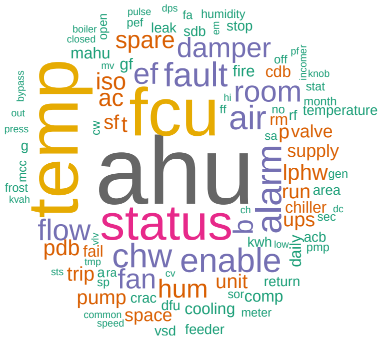


Occupancy Modelling

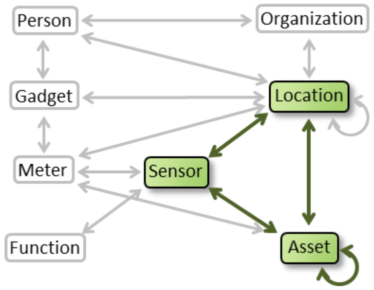


Schema comparison criteria

The meta-data schemata were compared based on the criteria:



Completeness: Ability to represent the distinct tags found in the analyzed datasets.



Coverage: Ability to encode the information dimensions and relationships needed for applications.

?

Flexibility: Ability to express uncertainty in the metadata (e.g. which set of lights is controlled), or new sensors (e.g. iBeacon) and applications (e.g. smart couch).

Semantic Sensor Networks

Definition

Targets information retrieval from global sensor networks. The SARAF ontologies maps common concepts of different building ontologies.

Complete (–)

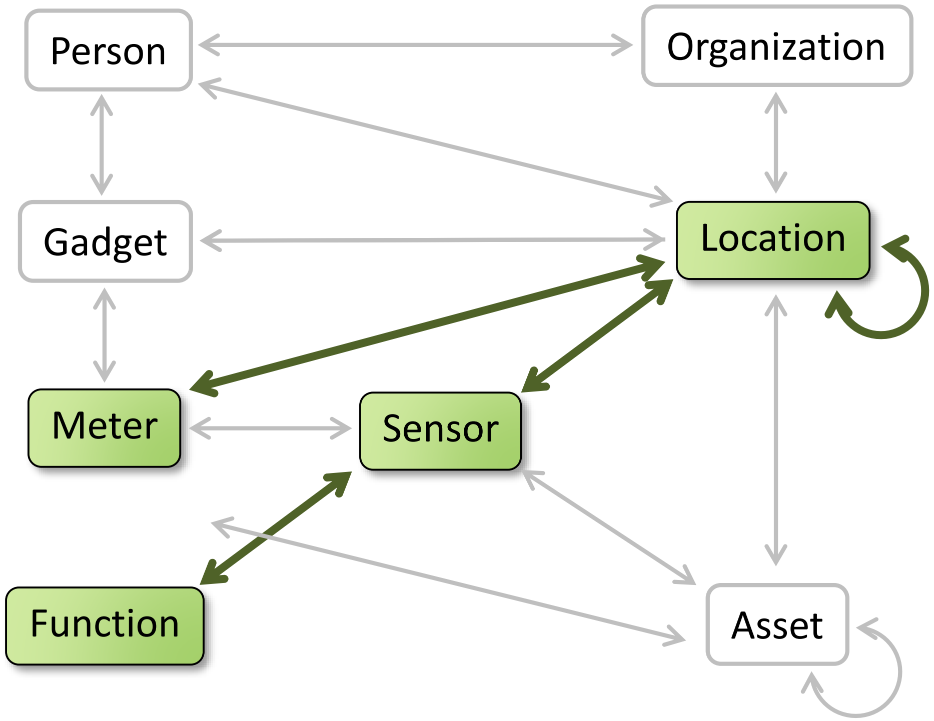
- – supports 11% of unique tag
- – supports 8% of weighted tags
- – only 5 basic sensors types

Coverage (/)

- + location hierarchy
- – no asset hierarchy
- / multiple references to external ontologies to model other dimensions

Flexibility (+)

- + ontologies are designed to be extensible
- + good separation of model and content
- – many different ontologies



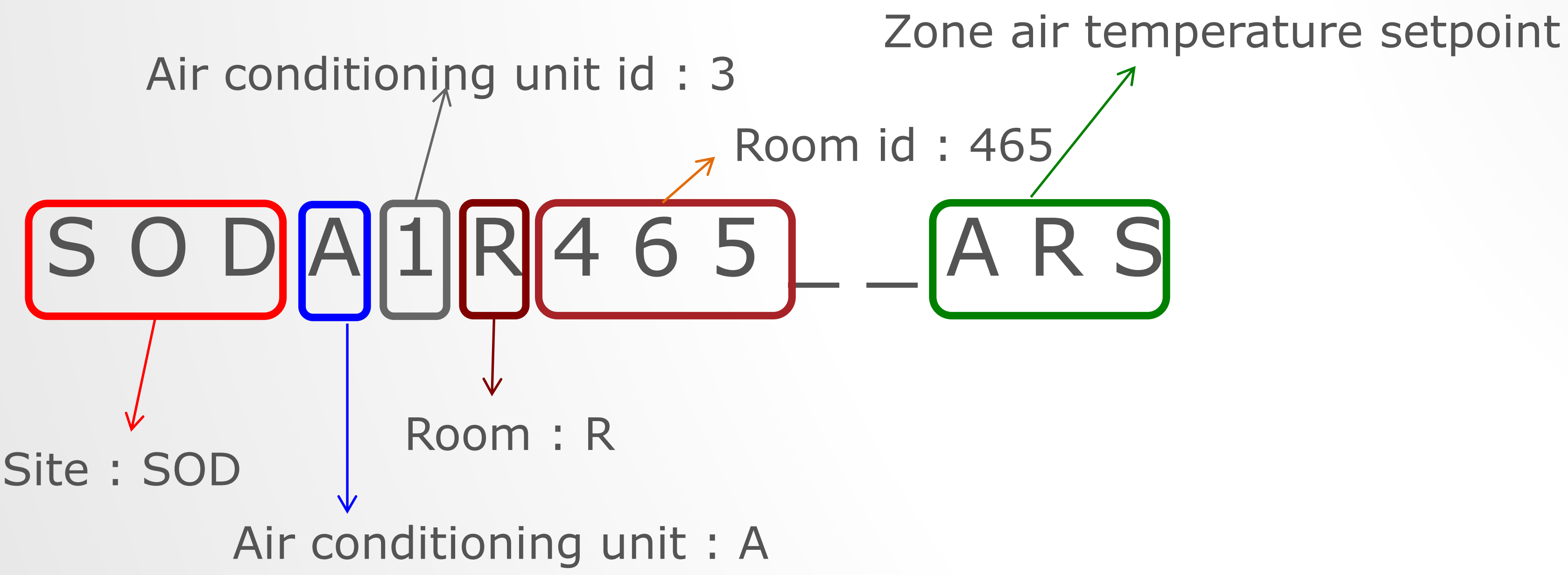
Coverage	Haystack	IFC	Semantic
Feedback / MPC / FDD	100%	100%	40%
Web Displays	75%	100%	100%
NILM / DR	50%	50%	50%
Energy Apportionment	57%	86%	57%
Occupancy Modelling	42%	58%	25%

Key findings

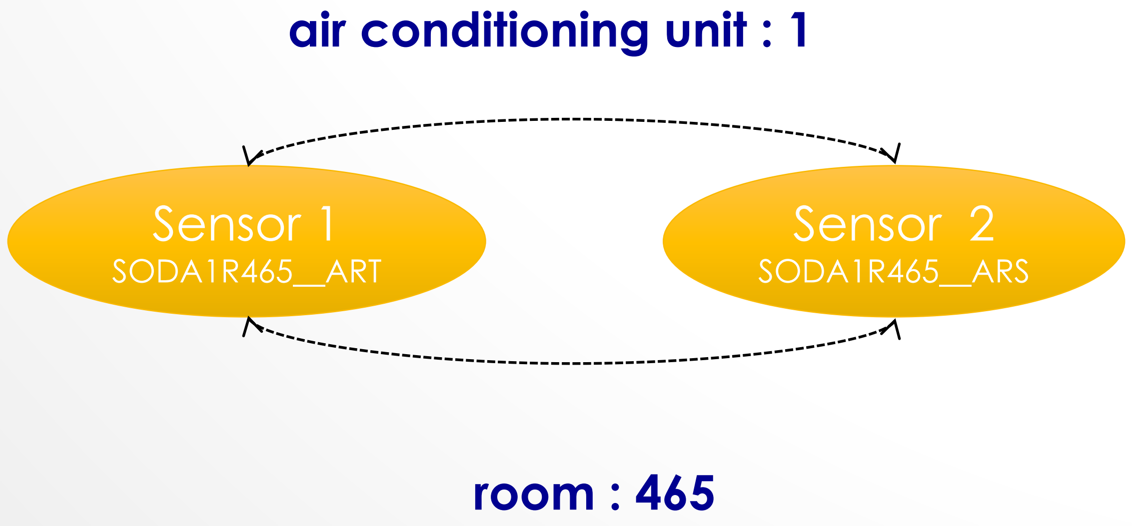
- The information contained in datapoint list is very diverse. Common, frequent tags exist, but, the composite tags are usually building specific.
- None of the meta-data schemata is complete or expressive enough. As long as this problem is not solved, BMS vendors will use their own schema.
- No existing metadata schema is flexible to capture novel sensors. This will in particular render the integration of rapidly developing IoT devices problematic.
- Semantic sensor web ontologies are too generic and fragmented to be of practical relevance. There need to be

  (a) a well-defined taxonomy of common building functions,
  (b) concepts for modelling building locations, assets, and persons,
  (c) tools which are easy to use by domain experts.

EXPERIENCE WITH (AND VALUE OF) PORTABLE APPS IN BUILDINGS - UCB



Application : Rogue Zones – A zone which is too hot or too cold



Example App: Finding Rogue Zones in Each Building

- Zones too hot or too cold
  - Leads to wastage of energy
- Indicates that energy is being wasted by air handling units.
- Impossible for building manager to continuously monitor 100s of zones in a building.
- A-priori installed building management systems generally do not install rogue zone detection.
- Requires finding “room temperature” and “room setpoint” sensors which are in the same room

App Pseudocode : Detecting Rogue Zones in Buildings

```
Zones = GetAllZones()
For each zoneid in Zones:
   CalculateRogueZone(zoneid)

CalculateRogueZone(zoneid):
      Temp_Sensor   = getZoneSensor(zoneid, "zone temp sensor")
      Temp_Setpoint = getZoneSetpoint(zoneid, "zone temp setpoint")
       if Temp_Sensor_t  - Temp_Setpoint_t  > threshold for all timesteps:
         return True

Example query against metadata store:
getZoneSensor(zoneid, "zone temp sensor")
   select sensor where zone-id=zoneid and sensortype="zone temp
sensor"
```

Other Applications

- **Finding Inefficient Air Handling Units :**
  - Air handling units may serve both over-heated and over-cooled zones
    - Indicates that cooling overheated zones is leading to over-cooling of other zones.
  - Leads to wastage of energy
  - Requires finding rooms which are served by the same air handling unit.
- **Identifying existence of night-time setbacks in buildings.**
  - Identifies if various components in a building run on the same schedule 24x7 or has setbacks
  - Absence of setbacks indicates easy opportunities for energy saving.

Results across 10 buildings

Building Id	Year of Construction	BMS Vendor	Num. of Sense Points	Num. of Thermal Zones	Num. of Hot Rogue Zones	Num. of over-cooled zones	Num. of AHUs	Num. of Inefficient AHUs
1	1994	1	1586	201	5	17	4	2
2	2009	2	2522	78	2	0	NA	NA
3	1961	1	367	42	28	1	2	1
4	1968	1	132	12	1	0	2	0
5	1941	1	417	48	8	4	6	2
6	2007	1	6169	368	35	5	NA	NA
7	NA	1	164	8	3	0	6	0
8	1950	1	421	20	0	2	1	0
9	1982	1	277	9	2	0	3	0
10	1996	1	730	57	10	0	1	0
Total			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

BACKUP

Project Haystack

Definition

Open-source initiative to define text labels to annotate datapoints

Complete

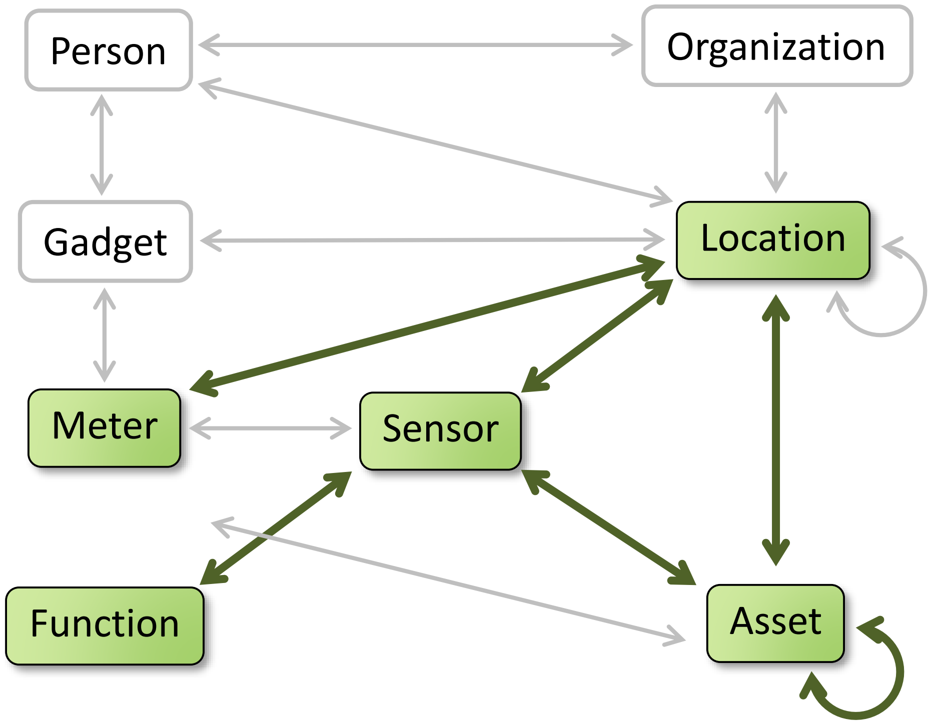
- + supports 54% of unique tag
- + supports 63% of weighted tags
- + common sensors types
- − few sensors beside the HVAC (e.g. light control)

Coverage

- + HVAC hierarchy
- − no location hierarchy

Flexibility

- + new text tags can be defined
- − no way to capture any uncertainty



Coverage	Haystack
Feedback / MPC / FDD	100%
Web Displays	75%
NILM / DR	50%
Energy Apportionment	57%
Occupancy Modelling	42%

IFC (BIM – Building Information Models)

Definition

Standard for information management in buildings. Originated from 3D CAD model exchange.

Complete

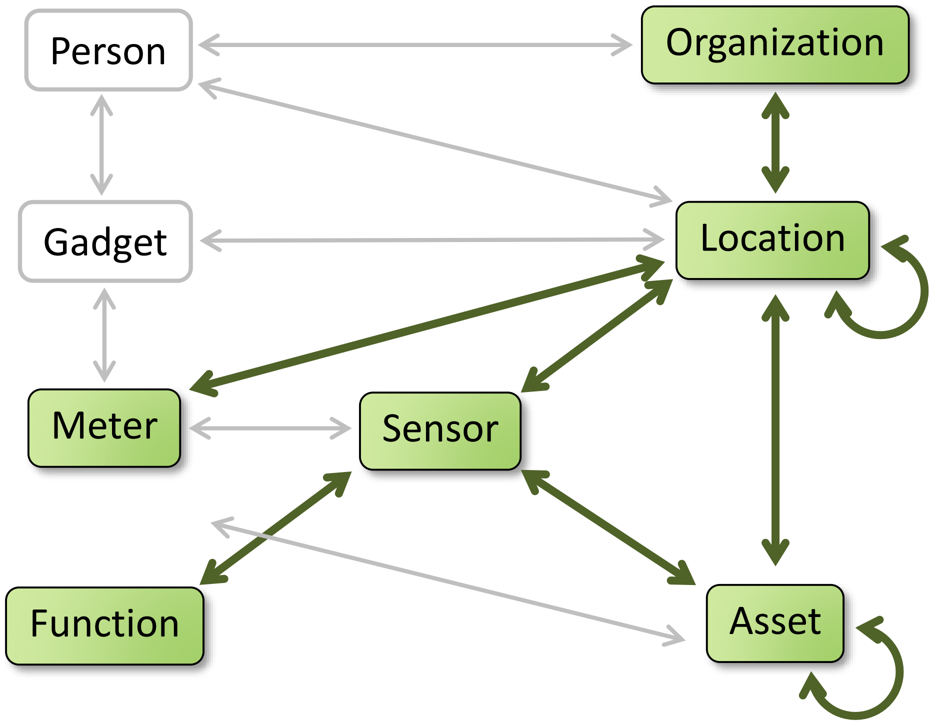
- supports 29% of unique tag
- supports 60% of weighted tags
- 11 very common sensors types

Coverage

- strong on location modelling
- strong on asset modelling
- model complexity complicates information retrieval

Flexibility

- format is extensible
- standardization new elements requires consensus of CAD community



Coverage	Haystack	IFC
Feedback / MPC / FDD	100%	100%
Web Displays	75%	100%
NILM / DR	50%	50%
Energy Apportionment	57%	86%
Occupancy Modelling	42%	58%