Metadata: (Real World) Example, Motivation, and Some Implications

WISHI Presentation
Feb 5, 2018

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

Outline

- Industrial-grade (building) example, issues

- Objectives
  - Annotation, context for apps/services (M2M?) understanding
  - (ultimate challenge) How to model complex systems?
    - e.g. building, production line, or factory floor

- Metadata example

- (some) Learnings and observations
  - Experiences from actual implementations and products

(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 semantic annotation
  - Point names assigned arbitrarily by installers/integrators, e.g. "zn3-wwfl4"
  - BMS control sequences are custom tailored using those names
    - Completely obscures structural relationships, layout of equipment
    - Each building looks different, even those using the same BMS

- Problems
  - Expensive, brittle, obscure, error prone, not scalable, prohibitive changes/additions
  - Cannot have portable apps and services, e.g. AI, analytics
    - Even across buildings with the same BMS, let alone vendors..

Sensor data and meta-data use

- Sensor "`zn3-wwf14`"" "`77.6`" ??

- Services, analytics, benefit from additional metadata info
  - Is a zone temperature
  - Is an exterior zone
  - Is South facing
  - 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"               //
```

Learnings and Observations

- (rich) Metadata needs to be handled in IoT systems
  - Lower cost, flexibility: flexibility to add sensors, new functions, e.g. predictive maintenance; market choices (vs, vendor single source)
  - Facilitate attribute-based search, a la SPARQL

- Common tag naming = pragmatic (almost) semantic substitute
  - Apps and services use common dictionary to "infer" meaning

- Descriptive, not prescriptive
  - Does not mandate which tags to use with which entity BUT
  - defines how to name and structure tags when used

- Metadata in IoT: many variations and combinations, rarely changing = require special treatment?
  - Opt1: allow variable metadata key, value pairs in info models
    - Does not fit well with fixed object structuring with metadata as prescribed properties
  - Opt2: separate APIs/queries to fetch metadata as an overlay?
  - Opt3: xxx?