One Data Model Weekly Teleconference Agenda and Content

May 23, 2022

Agenda

- Announcements & Admin
- Scott-Logan update
- Update on Sensor Type meta-model
- Action review
- AOB

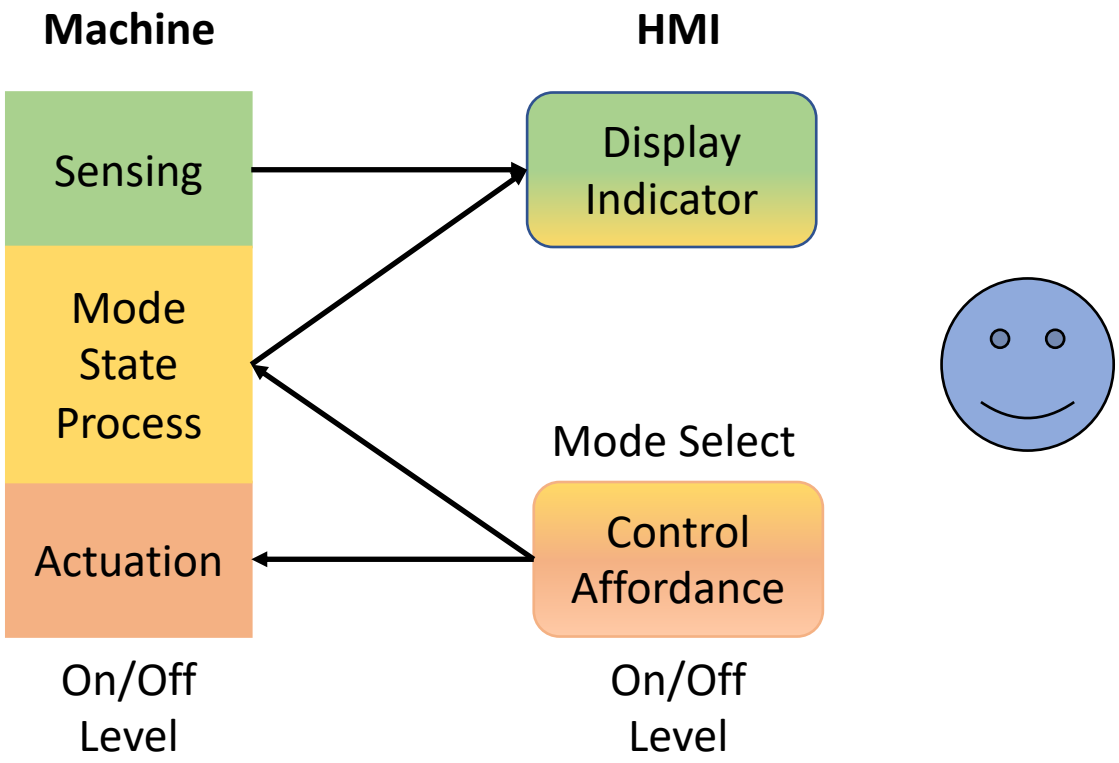
Announce

- WISHI working meeting May 19th

Types of Models (review from 5/9)

- Sensing elements (temperature, humidity, voltage)
- Control elements (on/off, level, actuation, display, HMI)
- Process, mode elements (mode, grinding, brewing, foaming)
- Domain purpose and complex control (lighting, medical, media, closures, energy (DRLC, microgrid), IAS, HVAC)
- Appliance component (Icemaker, Printer queue), often named status elements and standalone affordances

Sensing and Control



Sensor Meta-Model

- Develop a common abstraction for sensing devices
- Sensor intrinsic
  - Underlying physical dimensional model for quantities
  - Context (currentValue, setpoint, limit value, statistic)
  - State space occurrences, 1/t, summaries
  - Rectification of values into state space occurrences
  - Substance types and other Feature of Interest
- Value characterization
  - Unit
  - Scale (Value Range in unit) e.g. 0-100C
  - Quantization (Interval between represented values)

Sensor Meta-Model (2)

- How information is reported
  - Number type and value encoding
  - Data shape/ data schema
  - Reporting control, conditions
  - Aggregation and series reporting, rectangular formats
  - Network and application protocol binding
- Associated data
  - Timestamps
  - Geolocation
  - Data goodness

Extensions and Definitions

- Types and integration patterns
  - Dimensional and Derived Quantities: mass, length, time
  - Context types: minimum, maximum, presentValue
  - Substance types: CO2, Methane, Water
  - Feature of Interest for categories: Door, Window
  - Unit system
  - Optional: Timestamps, Geolocation, Data Quality
- Reporting
  - Data Shape, encoding formats
  - Format for aggregate data
  - Reporting control, time and threshold
  - Ecosystem binding

Action status

- MJK - PG models failing CI pointercheck
- MJK – Adoptable model progress

Closing

- AOB
- Next teleconference May 30, 2022

Back up

Strawman Model Selection

- Goal: 20-30 useful models without composition
- OCF Models
  - Component models (mode, foaming, brewing, grinder…)
  - Model groups (e.g. Medical devices, Media, Lighting)
  - Common Pattern Sensors (most useful models)
  - A few standalone models (windowcovering, TTS, PTZ, airquality)
- OMA Models
  - Common Pattern Sensors
  - Standalone (Buzzer, textdisplay, audioclip)
- BT Models
  - Generic, Lighting, Sensor, Scene, Time models
  - Servers, States and Properties, Characteristics (quantity/unit/size)
- Zigbee Models
  - Similar to OMA and OCF in scope of definitions

Types of Models

- Sensing elements (temperature, humidity, voltage)
- Control elements (on/off, level, actuation, display, HMI)
- Process, mode (mode, grinding, brewing, foaming)
- Domain purpose (lighting, medical, media, closure, energy, IAS, HVAC)
- Highly abstract appliance (Icemaker, Printer queue), usually named status elements and standalone affordances

Unique features

- OMA
  - Named objects for sensed quantity types
  - Common data item semantic definitions
  - Min/Max + operational Min/Max
- OCF
  - Named objects for sensed quantity types
  - Common sensor data items (some named values)
  - Range/step/precision
- Zigbee
  - Named objects for sensed types, composed devices
  - RPC based with fancy reads and writes and binary types
- Bluetooth Mesh
  - Semantic data model with Device Property specialization
  - "States" for control actions with own semantics level vs. power
  - Function groups and aggregate data reporting

Summary

- Common patterns and practices are needed, and are not embodied in the contributed or potential models
- Sensors, Actuators, and standalone abstract models are interesting as useful models to harmonize
- Sensor harmonization
  - Taxonomy for what is sensed and unit
  - Abstraction that includes use categories on one unit (presentValue, range, min/max, step, precision…)
  - Data type consistent with the physical embodiment
  - Wire formats as bindings and mappings
- Actuation harmonization
  - Simple actuation and HMI can be defined with common options
  - Complex actuation had domain dependencies
- Standalone models e.g. air quality just need to be vetted
- Domain models may be more difficult to harmonize

Class and Instance

- Class as template for an instance
- Semantic categorization
- Reuse vs. specialization tradeoff at each level
- Composition of classes as template for composed instances, etc.
- Refinements: Class, Sub-class, Type, Sub-Type, Kind
- Classes all the way down (?) to the final specialization
- "Library instances" are templates
- IP Address, unique identifier, cardinality of one
- All values filled in or nulled
- Has protocol bindings and ecosystem-specific settings

Agenda

- Announcements & Admin
- Action list
  - Document common patterns and usage – md document in a repo with issues – generic questions/problems
  - What we encountered in PG review/**convergence issues**
  - Alignment directives - https://github.com/one-data-model/onedm/wiki/OneDM-model-alignment-directives
  - Create a new repo with this as a section
  - Bluetooth Mesh models – fix the ones in playground
  - Zigbee models ""
  - Schema extension – cddl pattern?
- State of ASDF
  - ASDF PR #74
  - Other PRs in progress
- Classes, instances, and successive refinement

Action list

Action list

- Document common patterns and usage – md document in a repo with issues – generic questions/problems
- What we encountered in PG review/**convergence issues**
- Alignment directives - https://github.com/one-data-model/onedm/wiki/OneDM-model-alignment-directives
- Bluetooth Mesh models – fix the ones in playground
- Zigbee models ""
- Schema extension – cddl pattern?

Announce

- ASDF design team meeting wed. OneDM slot
- Digital Twin Consortium - standardize interfaces?
- T2TRG Digital Twin workshop early May 4th
  - Prelim agenda in next 2-3 days, then announce
  - OneDM Wed slot

Thing vs. Object

- Names: Object and Thing
- The building block aspect in OneDM
- Nesting (we already have thing in thing nesting)
- Objects and Properties (Events, Actions) at the same level

Playground CI errors

- Remove ZCL and BT models to a development branch

OCF models and OMA models

- Models are uniform between OCF and OMA
- OCF repo that auto-converts OMA models
- Supports the converter – name conversion to unique names
- OCF repo is a later version of the OCF database
- CI still needs some integration + automation
- Linter could return an error code
- Linter versions? Linter repo to clone from…

February 14th Agenda

- Announcements & Admin
- OneDM Update for T2TRG
  - March 10
  - 10 minutes + Q&A
- Extending to include ecosystem models
  - CI process to convert new SDF models to ecosystem
  - Goal of making OneDM models work for everyone
- Extend the playground to more ecosystems
  - Bluetooth Characteristics and data types
  - Other?
- Ecosystem-specific models
  - Namespace and versioning scheme

Ecosystem-specific models

- Namespace and versioning scheme
- Playground models don't have a namespace –
  - need to add the block
  - linter check and warning?
  - Is there a default behavior when there is no namespace reference? (research) no contribution to a global ([https://onedm) namespace
  - Can't export but can import using explicit prefix
  - These are theoretically ecosystem URIs (ocf:, oma:) or should they all be onedm – is there a useful constraint beyond "onedm:"
    - Ecosystem root, (~~github root~~), onedm root
    - onedm is preferred as a unifying principle
    - BSD3 strongly recommended and required for adoption
    - https://github.com/one-data-model/playground/wiki/Namespaces-and-versions-(2022-02-transition)
    - Please make a note in this wiki about what you will do with namespaces
    - Ari and Wouter will update pg models and wiki
- Versioning question
  - What is our resolution of the version marking question?
  - Discuss at next call February 21
  - Ecosystem-specific versioning - is a string good enough? Monotonicity?
  - What are the operations on ecosystem versions

February 7th Agenda

- Announcements
  - T2TRG meeting – 15m on agenda
  - DTDL conversion code is in the IPSO github
  - https://github.com/EricssonResearch/ipso-odm/tree/sdf-dtdl#onedm-sdf-to-dtdl-converter
- Bespoke repositories, OMA model repo
- Adoption process
- OneDM new venue development
  - Outreach on announcement of the next meeting
  - Groups.io delivery options
- AOB
  - Expanding the playground – BT models
  - Ecosystem-specific source representations in the playground (e.g. XML)(Back-translations of all PG models to an ecosystem format)

Bespoke Repositories

- IPSO and OCF repos created
  - Carsten get admin permission
- Instantiate a repo from template for each?
  - Need to try this
- Haven't designed the linker for separate namespaces – needs to be added – can be a simple file that is manually updated for now
- Write up a document and update existing documents to reflect the addition of repos
- Blog post introducing and announcing

Adoption process

- Status and plan
  - Original model proposal still needs closure on details
  - Updates will be needed -
  - Multiple drivers needed for models? OCF + IPSO
  - Adoption in 1+ steps with deferred convergence/alignment – less strict – phase
    - For example, ISO 8601 duration
    - Common abstraction with minimal constraints
    - DTDL example
    - How can we address un-solvable issues like conversion to/from duration in months – e.g. how do we import these models
    - Units are hard
    - We should only use pattern in targeting to a representation

Announcements

- New mailing list at groups.io
  - https://groups.io/onedm

Venue shift

- New mailing list at groups.io
  - https://groups.io/onedm
- Move enterprise admin, calendar, etc. to github
- Set up a blog on onedm.org

2022-01-31 conclusions

- Review board will not push for adoptions
- (Best practices evolve)
- Sensor design by a dedicated team – breakouts vs. agenda working time
- Ecosystem repo tracking

Review, Adoption, Publication

- Review from last meeting
- Alignment as a background process
- Toolchain
- Next steps, going forward
  - Get multiple spaces going as a priority
  - Announce this on a blog article onedm website
  - Monthly publications, news and updates
- People can run private repos on their own account with onedm CI as github actions
- Public models on onedm.org
  - Template work task
  - Wouter and Carsten can work from a script (run.sh in the pg)
  - Github platform installed packages can easily be resolved
  - Orgname in the repo naming e.g. ipso-models ocf-models sunspec-models
- Start with (ipso or oma) –models
- Bluetooth mesh? Practical considerations need characteristics and data types, need a converter – need URI for the source so translations can be automated – start by working with Simon on publishing the underlying dictionaries

Venue changes

- Teleconferences
  - TBA new teleconference venue
  - Same schedule
- Mailing list
  - Planning to move to groups.io
- Github for everything else
  - Calendar
  - Work planning
  - OneDM Causeway data

Alignment

- First pass – names, SDF usage
- Who needs to change what and why
- 2 levels of bridge
  - Same meta-model – SDF + structural isomorphism
  - Same information model – OneDM Adopted per function
- Translation across ecosystems using 1:1 translators without SDF is required if there are no SDF models
- Benefit is in adopting SDF models from other ecosystems
- Benefit in having models published under ecosystem "banner" to facilitate cross-enrichment
- Minimal alignment opportunities?
- OCF has collections – how do we model this in SDF? SDF Thing
- IPSO models compose differently – Object links with SDF Thing
- Atomic collections need one read for multiple objects + required composite read in OMA LWM2M

Alignment

- Separate folder/repo for pre-alignment
- No name conflicts due to prefix – same as separate repo
- What are common improvements that can be carried forward for alignment?
- Quick win in what we can currently translate in our own repos and how to use them – how to translate identifiers to rt, how to combine OCF interfaces, etc.
  - What if each OCF interface is a combine-able object that can be used in runtime systems?
- Can use prefixes in addition to separate governance as a quick win?
- Can use URN naming to identify which model ecosystem the models come from
- Building a bigger ecosystem of models
- Can do the unifying thing as a separate thread

Toolchain issue for creating new repos

- Can be as simple as creating more repos in OneDM
- Tooling needs to work across repos
- Github actions integration
- Recipe for the existing tools
- Clone a repo with a template

Going forward on multiple repos

- Sketch out solution and sort out on mailing list
- New mailing list – gittr?

Review Board

- Review the discussion from the week of 9/13
  - We would rather focus on function and architecture and leave naming to later in the process
  - We should prioritize developing our way of working in the first models
  - On/off switch is not so simple due to the high degree of generality and reuse; this may be the case with many common affordance types
  - Look at a few more examples and continue to review the sensor patterns – start a discussion in wiki(HackMD) => issues => PR to track the discussion
    - Units
    - Review isomorphism in existing models, what are the common elements and what is a common set of design patterns (inheritance, mixins, modularity) – common factors, leverage someone's IPSO background - sdfRef with multiple sources

Next Steps

- Models have been reviewed – need to process the review comments and proceed with the discussion
  - Can these models be provisionally accepted with the expectation that they can be refactored as we gain more experience?
  - More guidelines may be needed, what about looking at a broader set of models to see if this is the case?
  - Monday review board to further progress the models we have
- Ari has made a PR to the new repo

Review Board

- Discussed versioning into the repo – semantic version is added when a contribution reaches a certain state
- Discuss CLA signup in github – no legal org? agreement to use BSD3 is sufficient - where is this done? It is a step in submitting PR to check that the submitter has agreed to the CLA
  - Github policy is a blanket agreement where the repo carries a license:
  - https://docs.github.com/en/github/site-policy/github-terms-of-service#6-contributions-under-repository-license
- Not going to block contributions
- Note on making a private development fork+branch for contributions, ~~how to send notifications using an issue~~
  - Github has a draft PR process we can use, so we can make a PR in the fork

SDF Interim summary

- No new additions
  - not including sdfProduct
- Double check with OneDM users that SDF has the required features
  - Statements from organizations that will publish models, etc.
- Please review the latest draft

OneDM Roadmap

- SDF is almost done and on a track for evolution
- OneDM has developed SDF tools and a governance regime, playground and examples, and a good start on best practices
- Other JSON Schema based data models are proliferating in both single- and multi-domain groups
- iotschema.org is not active
- W3C Web of Things needs a semantic annotation system

How to make Progress

- Determine what kind of models are needed
- Keep the model discussion at the high level
- Use cases vs. requirements and scenarios

Sensor design pattern

- Separate the quantity from the sensing
- "Temperature Sensor" vs "Sensor" that measures "Temperature"
- Make it easy for many embodiments to be constructed

Data

- Fundamental types
- Quantity and Units
- Characteristics (i.e. averages, quartiles, etc.)
- Range and Scale
- Reusability of these definitions
- Data type (sdfData) vs sdfProperty, where do the application semantics like quantity and units go?
- Examples of different styles, use common quantities temperature (pressure is differential, gauge vs. absolute) how do we differentiate?
- Data should be standalone

Interactions

- Functional modularity
- Actions and Events could be optional and bespoke for certain protocols
- The base model could be very simple with well categorized extension points e.g. Min/Max observed
- Sensors are separate from generic Properties, sensor definitions use property definitions and add behaviors like conditional reporting
- Properties can be used for static or dynamic
- Features? Modular composition

Semantic Composability

- Modularity
- Conceptual Layering
- Composed Properties
- Syntactic considerations and structural implications for models, e.g. mixin vs collections
- Mixin should have optionality and satisfy the LCS principle
- Multiple "inheritance" requires conflict resolution
- Ordering dependencies, need to prototype

Protocol dependencies mapping

- Data reporting options including composed schemas, sequences, conditional notification
- Read/write/observability
- Late binding representations
- Mapping files add information to the model where the information comes from external namespaces. Namespaces are part of the map key and want to become segments in the JSON pointer
- Need to manage cross-namespace references

- HackMD examples
- SDF examples also, test against the current specification