

P1752 2nd Meeting on Schema Review

Sponsored by IEEE Engineering in Medicine & Biology (EMB) Standards Committee

14 and 17 January 2020

Teleconference

Intro to Schemas

- A schema defines how instance data should look like in terms of data types, properties, etc.
- P1752 schemas are written in JSON Schema <https://json-schema.org/>
- Documentation available here <https://json-schema.org/understanding-json-schema/index.html>
- To get familiar with schemas and data, you can check <https://www.openmhealth.org/documentation/#/schema-docs/schema-library>
- The reference above is an om-ramp to understating schemas
- It allows viewing schemas and sample data on the same page, without having to download files, and all the pages are public.

“Utility” schemas

- Draft standard schemas stored in iMeet
<https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyNDgxNTY2XQ/>
- 1. Units of measure (e.g., length, duration)
- 2. Value sets (e.g., body posture, descriptive statistics)
- 3. Time-related schemas (date-time, time interval, time frame)
- Defined once, referenced by other schemas, as needed
- → not all units of measure or value sets need to be defined in a separate schema

1: Units of measure

- unit-value.json is a generic schema
- It defines an object with two properties
 1. value of type *number*
 2. unit of type *string*
- It also requires both value and unit for data to be compliant

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "description": "This schema represents a numerical value with a unit of measure.",
  "type": "object",
  "properties": {
    "value": {
      "description": "The numeric value of the element.",
      "type": "number" ←
    },
    "unit": {
      "references": [
        {
          "description": "The unit of measure of the element. Allowed values are drawn from the Common synonyms (non-UCUM) column of [subset of] UCUM, SI and English units. ",
          "url": "http://download.hl7.de/documents/ucum/ucumdata.html"
        }
      ],
      "type": "string" ←
    }
  },
  "required": [ ←
    "value",
    "unit"
  ]
}

```

Duration unit value

- duration-unit-value schema references the unit-value schema
- it inherits properties and requirements
- adds a restriction on the unit property via a value set
- the reference section describes how the value set was chosen

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "description": "This schema represents a duration or length of time.",
  "type": "object",
  "allOf": [
    {
      "$ref": "unit-value-1.x.json",
    },
    {
      "properties": {
        "unit": {
          "references": [
            {
              "description": "The unit of measure of the element. Basic unit is second (s). Allowed values
are drawn from the Time Units Common Synonyms (non-UCUM). The valid UCUM code is different for second (s), month (mo) and
year (a).",
              "url": "http://download.hl7.de/documents/ucum/ucumdata.html",
            }
          ],
        },
      },
      "enum": [
        "ps",
        "ns",
        "us",
        "ms",
        "sec",
        "min",
        "h",
        "d",
        "wk",
        "Mo",
        "yr"
      ]
    }
  ]
}

```

2: Value sets

- previous slide has an example of a value set defined within a schema (another example on the slide after the next one)
- some schemas are just value sets
- a value set is an array of possible values (enumerated list)
- the reference section describes how the value set was chosen
- data must include one of the allowed values to be compliant
- may be exhaustive or not (in which case they can be expanded)


```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "string",
  "description": "The posture of the subject (for example, during a measurement). This value set is not exhaustive.",
  "references": [
    {
      "description": "The SNOMED code represents Position of body and posture (observable entity)",
      "url": "http://purl.bioontology.org/ontology/SNOMEDCT/271605009"
    },
    {
      "value": "sitting",
      "description": "The SNOMED code represents Sitting position (finding)",
      "url": "http://purl.bioontology.org/ontology/SNOMEDCT/33586001"
    },
    {
      "value": "lying down",
      "description": "The SNOMED code represents Lying (Recumbent) position (finding)",
      "url": "http://purl.bioontology.org/ontology/SNOMEDCT/102538003"
    },
    {
      "value": "standing",
      "description": "The SNOMED code represents Standing position (finding)",
      "url": "http://purl.bioontology.org/ontology/SNOMEDCT/10904000"
    },
    {
      "value": "semi-recumbent",
      "description": "The SNOMED code represents Semi-recumbent position (finding)",
      "url": "http://purl.bioontology.org/ontology/SNOMEDCT/272580008"
    }
  ],
  "enum": [
    "sitting",
    "lying down",
    "standing",
    "semi-recumbent"
  ]
}

```

physical-activity.json

```
  "reported_activity_intensity": {  
    "description": "Self-reported intensity of the activity performed.",  
    "references": [  
      {  
        "description": "WHO: What is Moderate-intensity and Vigorous-intensity Physical Activity?",  
        "url": "https://www.who.int/dietphysicalactivity/physical\_activity\_intensity/en/"  
      },  
      {  
        "description": "US Office of Disease Prevention and Health Promotion: Top 10 Things to Know About the  
Second Edition of the Physical Activity Guidelines for Americans",  
        "url": "https://health.gov/paguidelines/second-edition/10things/"  
      }  
    ],  
    "type": "string",  
    "enum": [  
      "light",  
      "moderate",  
      "vigorous"  
    ]  
  },
```

```

"$schema": "http://json-schema.org/draft-07/schema#",

"description": "This schema represents episode(s) of physical activity.",
"type": "object",

"definitions": {...},

"properties": {
  "activity_name": {
    "description": "Name of the activity. This can be sedentary.",
    "$ref": "#/definitions/activity_name"
  },
  "base_movement_quantity": {
    "description": "Number of repetitions of the activity base movement, if applicable (e.g., if activity is walking, base_movement_quantity would be
    "allof": [
      {
        "$ref": "#/definitions/unit_value"
      },
      {
        "properties": {
          "unit": {
            "anyOf": [
              {
                "enum": [
                  "steps",
                  "laps",
                  "skips",
                  "movements",
                  "strokes"
                ]
              },
              {
                "type": "string"
              }
            ]
          }
        }
      }
    ]
  },
  "effective_time_frame": {
    "description": "As a measure of a duration, physical activity should not be associated to a date-time time frame. Hence, effective time frame is
    "allof": [
      {
        "$ref": "#/definitions/time_frame"
      },
      {
        "required": [
          "time_interval"
        ]
      }
    ]
  }
}

```

3: Modeling of time

- Date time
- Time interval
- Time frame = date time OR time interval (used to define property *effective time frame*, used in all quantitative schemas)

on iMeet

time schemas <https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyNDgxNjExXQ>

total sleep time <https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyMzY0Mjk3XQ/>

ambient temperature <https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyMzY0MjY1XQ>

Date time

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "description": "This schema represents a point in time (ISO8601). If a timezone is not included, the timezone is assumed to be UTC.",
  "type": "string",
  "references": [
    {
      "description": "Reference RFC 3339 5.6 for details.",
      "url": "http://tools.ietf.org/html/rfc3339#section-5.6"
    }
  ],
  "format": "date-time" ←
}
```

Time interval

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "description": "This schema describes an interval of time. No commitments are made as to whether the start or end time point itself is included in the interval (i.e., whether the defined interval includes the boundary point(s) or not).",
  "type": "object",
  "definitions": {
    "date_time": {
      "$ref": "date-time-1.x.json"
    },
    "duration-unit-value": {
      "$ref": "duration-unit-value-1.x.json"
    }
  },
  "oneOf": [ ←
    {
      "properties": {
        "start_date_time": {
          "$ref": "#/definitions/date_time"
        },
        "duration": {
          "$ref": "#/definitions/duration-unit-value"
        }
      },
      "required": ["start_date_time", "duration"]
    },
    {
      "properties": {
        "end_date_time": {
          "$ref": "#/definitions/date_time"
        },
        "duration": {
          "$ref": "#/definitions/duration-unit-value"
        }
      },
      "required": ["end_date_time", "duration"]
    },
    {
      "properties": {
        "start_date_time": {
          "$ref": "#/definitions/date_time"
        },
        "end_date_time": {
          "$ref": "#/definitions/date_time"
        }
      },
      "required": ["start_date_time", "end_date_time"]
    }
  ]
}
```

Time frame

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "description": "This schema describes a time frame as a point in time or a time interval.",
  "type": "object",
  "definitions": {
    "date_time": {
      "$ref": "date-time-1.x.json"
    },
    "time_interval": {
      "$ref": "time-interval-1.x.json"
    }
  },
  "oneOf": [
    {
      "properties": {
        "date_time": {
          "$ref": "#/definitions/date_time"
        }
      },
      "required": [ "date_time" ]
    },
    {
      "properties": {
        "time_interval": {
          "$ref": "#/definitions/time_interval"
        }
      },
      "required": [ "time_interval" ]
    }
  ]
}
```

Example: Physical activity (PA) schema

- Schemas referenced
 - duration unit value
 - length unit value
 - time frame

- Physical activity

<https://iee-sa.imeetcentral.com/omh/folder/WzlwLDEyMjUyNTc4XQ>

PA Sample data

```
{
  "activity_name": "Running",
  "effective_time_frame": {
    "time_interval": {
      "start_date_time": "2019-03-29T08:26:03Z",
      "end_date_time": "2019-03-29T09:14:41Z"
    }
  },
  "distance": {
    "value": 7.45,
    "unit": "km"
  },
  "duration": {
    "value": 45.5,
    "unit": "min"
  },
  "kcal_burned": {
    "value": 383,
    "unit": "kcal"
  },
  "average_cadence": {
    "value": 184,
    "unit": "steps/min"
  },
  "cumulative_elevation_gain": {
    "value": 108,
    "unit": "m"
  },
  "duration_moderate_activity": {
    "value": 41.5,
    "unit": "min"
  },
  "duration_vigorous_activity": {
    "value": 4,
    "unit": "min"
  }
}
```

Example: Total sleep time

- Schemas referenced
 - duration unit value
 - time frame
- Total sleep time

<https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyMzY0Mjk3XQ/>

Total Sleep Time (details)

```
1  {
2  "$schema": "http://json-schema.org/draft-07/schema#",
3  "type": "object",
4  "description": "This schema represents total sleep time, i.e. The total sleep time is the interval between initial sleep onset time and final awakening time minus the duration
5  "references": [
6  {
7    "description": "The SNOMED code represents Duration of sleep (observable entity)",
8    "url": "http://purl.bioontology.org/ontology/SNOMEDCT/248263006"
9  }
10 ],
11 "definitions": {
12   "duration_unit_value": {
13     "$ref": "duration-unit-value-1.x.json"
14   },
15   "time_frame": {
16     "$ref": "time-frame-1.x.json"
17   },
18   "descriptive_statistic": {
19     "$ref": "descriptive-statistic-1.x.json"
20   },
21   "descriptive_statistic_denominator": {
22     "$ref": "descriptive-statistic-denominator-1.x.json"
23   }
24 },
25 "properties": {
26   "total_sleep_time": {
27     "description": "Total time asleep from bedtime until getting out of bed in the morning or across the 24-h period. This excludes any time that a person is awake after first
28     "$ref": "#/definitions/duration_unit_value"
29   },
30   "sleep_events": {
31     "description": "Individual sleep events and their durations to describe at what points throughout the night is the individual is asleep, and when summarized equal the total
32     "type": "array",
33     "allOf": [
34       {
35         "$ref": "#/definitions/time_frame"
36       },
37       {
38         "required": [
39           "time_interval"
40         ]
41       }
42     ]
43   },
44   "effective_time_frame": {
```

Total Sleep Time (cont'd)

```
44     "effective_time_frame": {
45       "description": "As a measure of a duration, time asleep should not be associated to a date time time frame. Hence, effective time frame is restricted to be a time interval.",
46       "allOf": [
47         {
48           "$ref": "#/definitions/time_frame"
49         },
50         {
51           "required": [
52             "time_interval"
53           ]
54         }
55       ]
56     },
57     "descriptive_statistic": {
58       "$ref": "#/definitions/descriptive_statistic"
59     },
60     "descriptive_statistic_denominator": {
61       "anyOf": [
62         {
63           "$ref": "#/definitions/descriptive_statistic_denominator"
64         },
65         {
66           "description": "If the value needed is a standard unit of duration, select from the duration-unit-value value set.",
67           "type": "string"
68         }
69       ]
70     }
71   },
72   "required": [
73     "total_sleep_time",
74     "effective_time_frame"
75   ]
76 }
```

Total Sleep Time (TST) Data

```
=====
total sleep time sample data: instance #1
=====
```

```
{
  "total_sleep_time": {
    "value": 5.5,
    "unit": "h"
  },
  "effective_time_frame": {
    "time_interval": {
      "start_date_time": "2019-02-19T22:30:00Z",
      "end_date_time": "2019-02-20T04:50:00Z"
    }
  }
}
```

```
=====
```

To review schemas on iMeet

1. Download sample data, schema and spreadsheet
2. Use text editor (e.g. notepad++ on PC, CotEditor on Mac, or similar applications that shows line numbers) to open sample data file to see how the data looks when modeled using the schema
3. Open and review the schema (property name and description, units of measure, required vs optional property, completeness of model, etc.)
4. Enter you comments into the spreadsheet (make sure to fill out column A,B,C,D)
5. Upload the modified spreadsheet to the iMeet (same location)
6. See also my emails on this topic

Recap

- A schema defines how data should look like in terms of:
 - data type (number, string, object, array, etc.)
 - properties (which ones, required vs. not)
- A schema may reference other schemas and if so, the requirements of the referenced schemas apply (e.g., unit-value requires a number + a string)
- Validation checks that instance data matches the template (data type of all properties, required properties, values from value sets, etc.)
- Some constraints can be defined in JSON Schema (e.g., number must be non-zero, string must match a pattern, etc.)
- Additional logical constraints are at the implementation level

Comments & Suggestions

- Say it now or send email to simona.carini@ucsf.edu