P1752 Meeting on Schema Review

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

- 10 and 13 December 2019
- 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/](https://json-schema.org/)
• Documentation available here [https://json-schema.org/understanding-json-schema/index.html](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](https://www.openmhealth.org/documentation/#/schema-docs/schema-library)
• The reference above allows you to view schemas and sample data on the same page, without having to download files, and all the pages are public.
Example #1: Duration unit value

- **Unit value**
  [https://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas/omh_unit-value](https://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas/omh_unit-value)

- **Duration unit value**
  [https://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas/omh_duration-unit-value](https://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas/omh_duration-unit-value)

- **Unit schemas on iMeet**
  [https://ieee-sa.imeetcentral.com/omh/folder/WzIwLDEyNDgxNTY3XQ](https://ieee-sa.imeetcentral.com/omh/folder/WzIwLDEyNDgxNTY3XQ)
Example #2: Modeling of time

- **Time interval**
  
  https://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas/omh_time-interval

- **Date time**
  
  https://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas/omh_date-time

- **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
Example #3: Total sleep time

- Schema referenced
  - duration unit value
  - time frame
- Total sleep time
  
  https://ieee-sa.imeetcentral.com/omh/folder/WzIwLDEyMzY0Mjk3XQ/
Total Sleep Time (TST) Data (1)

```json
{  
    "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"  
        }  
    }  
}
```
Total Sleep Time (details)
Total Sleep Time (cont’d)

"effective_time_frame": {
  "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."
  "allOf": [
    {
      "$ref": "#/definitions/time_frame"
    },
    {
      "required": [
        "time_interval"
      ]
    }
  ],
  "descriptive_statistic": {
    "$ref": "#/definitions/descriptive_statistic"
  }
},
  "descriptive_statistic_denominator": {
    "anyOf": [
      {
        "$ref": "#/definitions/descriptive_statistic_denominator"
      },
      {
        "description": "If the value needed is a standard unit of duration, select from the duration-unit-value value set."
      }
    ],
    "required": [
      "total_sleep_time",
      "effective_time_frame"
    ]
  }
}
Total Sleep Time Data (2)

```json
{
  "total_sleep_time": {
    "value": 330,
    "unit": "min"
  },
  "sleep_events": [
    "time_interval": {
      "start_date_time": "2019-02-19T22:30:00Z",
      "end_date_time": "2019-02-19T23:50:00Z"
    },
    "time_interval": {
      "start_date_time": "2019-02-20T00:15:00Z",
      "end_date_time": "2019-02-20T02:15:00Z"
    },
    "time_interval": {
      "start_date_time": "2019-02-20T02:30:00Z",
      "end_date_time": "2019-02-20T04:00:00Z"
    },
    "time_interval": {
      "start_date_time": "2019-02-20T04:10:00Z",
      "end_date_time": "2019-02-20T04:50:00Z"
    }
  ],
  "effective_time_frame": {
    "time_interval": {
      "start_date_time": "2019-02-19T22:30:00Z",
      "end_date_time": "2019-02-20T04:50:00Z"
    }
  }
}
```
Total Sleep Time Data (3)

descriptive statistic and descriptive statistic denominator schemas on iMeet
https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyNDgxNjEzXQ
Suggested review approach (see my emails)

1. Download sample data, schema and spreadsheet

2. Download and use tool (e.g. notepad++ or another simple text editor 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. Putting it all together:  
   https://ieee-sa.imeetcentral.com/omh/folder/WzlwLDEyMzY0Mjk3XQ/
Example #4: Physical activity (PA) schema

- Physical activity
  https://ieee-sa.imeetcentral.com/omh/folder/WzIwLDEyMjUyNTc4XQ
"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"}
This schema represents episode(s) of physical activity.

- **activity_name**: Name of the activity. This can be sedentary.
- **base_movement_quantity**: Number of repetitions of the activity base movement, if applicable (e.g., if activity is walking, base_movement_quantity would be the number of steps).
- **time_frame**: As a measure of a duration, physical activity should not be associated to a date–time time frame. Hence, effective time frame is

  ```json
  "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": [
      {"$ref": "#/definitions/time_frame"},
      {"required": [{"time_interval"}]
    }
  },
```
"distance": {
  "description": "The distance covered, if applicable.",
  "$ref": "#/definitions/length_unit_value"
},
"cumulative_elevation_gain": {
  "description": "The total ascent, if applicable.",
  "$ref": "#/definitions/length_unit_value"
},
"duration": {
  "description": "The net duration of the activity.",
  "$ref": "#/definitions/duration_unit_value"
},
"duration_light_activity": {
  "description": "The duration of light-intensity physical activity.",
  "$ref": "#/definitions/duration_unit_value"
},
"duration Moderate_activity": {
  "description": "The duration of moderate-intensity physical activity.",
  "$ref": "#/definitions/duration_unit_value"
},
"duration vigorous_activity": {
  "description": "The duration of vigorous-intensity physical activity.",
  "$ref": "#/definitions/duration_unit_value"
},
"average_cadence": {
  "description": "The average rate at which the activity was performed. If none of the units listed",
  "allOf": [{
    "$ref": "#/definitions/unit_value"
  },
  "properties": {
    "unit": {
      "anyOf": [{
        "enum": [
          "steps/min",
          "strokes/min",
          "skips/min",
          "laps/min",
          "movements/min",
          "strokes/lap"
        ],
        "type": "string"
      }
    }
  }]
}
"kcal_burned": {
    "description": "The calories burned during the activity.",
    "$ref": "#/definitions/kcal_unit_value"
},
"reported_activity_intensity": {
    "description": "Self-reported intensity of the activity performed.",
    "type": "string",
    "enum": [
        "light",
        "moderate",
        "vigorous"
    ]
},
"met_value": {
    "description": "Average Metabolic Equivalent of Task value for the activity",
    "type": "number"
},
"descriptive_statistic": {
    "description": "The descriptive statistic of a set of measurements (e.g., average, maximum) within
    "$ref": "#/definitions/descriptive_statistic"
},
"descriptive_statistic_denominator": {
    "description": "The denominator of the descriptive statistic when the measure has an implicit duration",
    "anyOf": [
        {
            "$ref": "#/definitions/descriptive_statistic_denominator"
        },
        {
            "description": "If none of the units listed applies, the string alternative will allow values
            "type": "string"
        }
    ]
},
"required": [
    "activity_name",
    "effective_time_frame"
]
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