A Simple JSON Format for ODM Definitions

Michael Koster

April 26, 2019

Overview

- Simple definition format for the ODM ontology
  - Object
  - Event, Action, Property
  - Data types
- Usable by domain experts
- Defined namespaces using curie notation
- JSON format description document

https://github.com/mjkoster/ODM-Examples/blob/master/sdf.md

- JSON Schema for validation and conversion

https://github.com/mjkoster/ODM-Examples/blob/master/sdf-schema.json

Simple Definition Format

```
{
  "info": {
    "title": "Example file for ODM Simple JSON Definition Format",
    "version": "20190404",
    "copyright": "Copyright 2019 Example Corp. All rights reserved.",
    "license": "http://example.com/license"
  },
  "namespace": {
    "st": "http://example.com/capability/odm"
  },
  "defaultNamespace": "st",
  "odmObject": {
    "Switch": {
      "odmProperty": {
        "value": {
          "type": "string",
          "enum": ["on", "off"]
        }
      },
      "odmAction": {
        "on": {},
        "off": {}
      }
    }
  }
}
```

Simple example – Info and namespace definitions

keywords

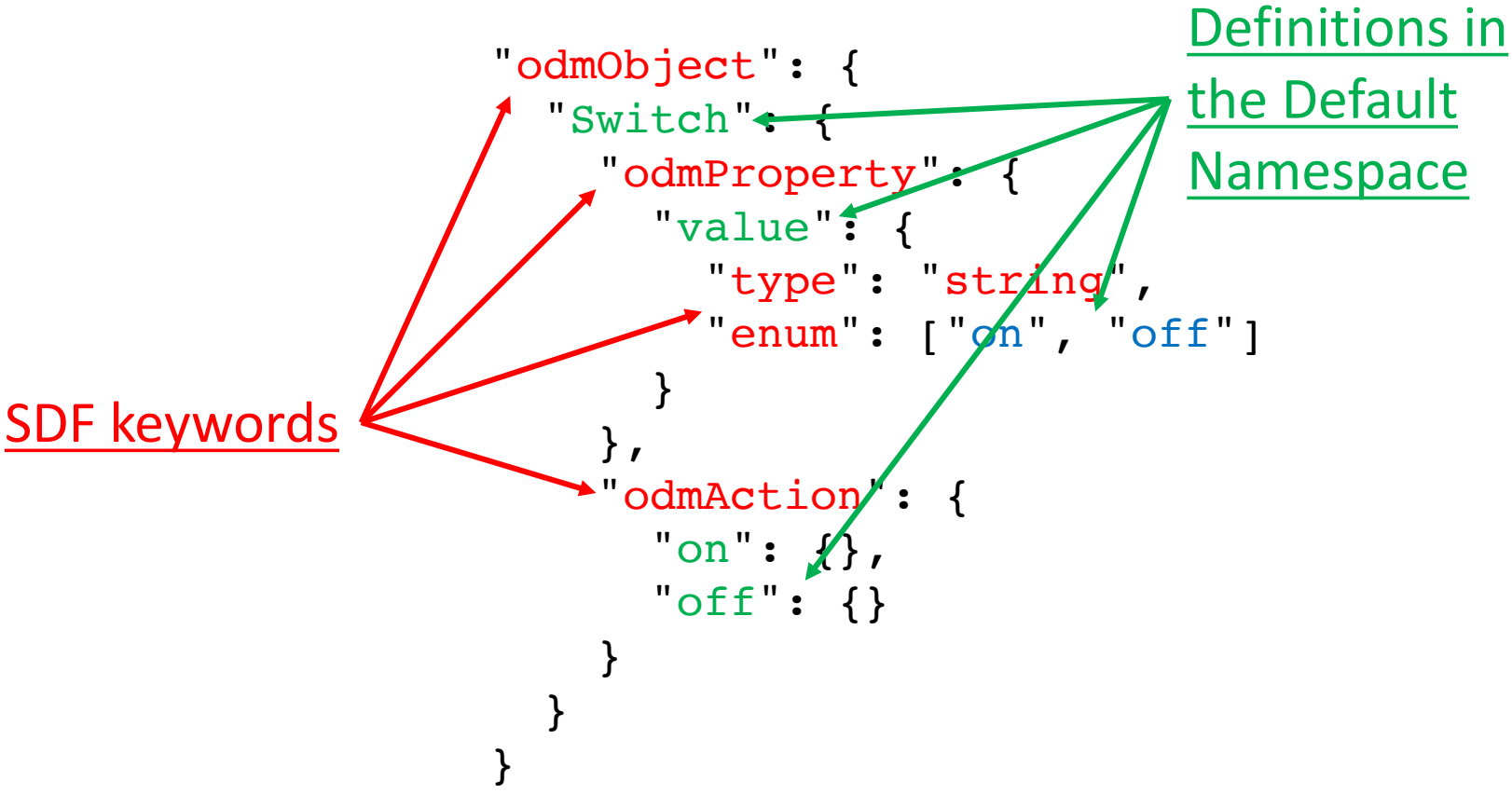
File Information

```
"info": {
  "title": "Example file for ODM Simple JSON Definition Format",
  "version": "20190404",
  "copyright": "Copyright 2019 Xcorp, Inc. All rights reserved.",
  "license": "http://example.com/license"
},
```

curies resolved

```
"namespace": {
  "ocf": "http://example.org/ocf/odm",
  "st": "http://example.com/capability/odm"
},
"defaultNamespace": "st",
```

Definitions



Definitions

- A definition consists of a defined term and a map of it's defined qualities

```
"value": {
  "type": "string",
  "enum": ["on", "off"]
}
```

Identifier scope and namespace resolution

1. Identifiers with an explicit namespace prefix
2. Keywords and quality names in the ODM and JSON Schema namespaces, defined in the JSON Schema for SDF at: https://github.com/mjkoster/ODM-Examples/blob/master/sdf-schema.json
3. Identifier defined in the same (local) block
4. Identifier defined in the next closest enclosing block recursively
5. Identifier defined in the file
6. Identifier defined in the default namespace

Tools

- Python for offline tools
- Nodejs for server tools
- Shell utilities
- Validation
- Linter
- Conversion utilities
- Database + lookup

```
info {
  title "Example file for ODM Simple JSON Definition Format"
  version "20190424"
  copyright "Copyright 2019 Example Corp.
        All rights reserved."
  license http://example.com/license
}
namespace {
  st http://example.com/capability/odm
}
defaultNamespace st
odmObject {
  Switch {
    odmProperty {
      value {
        type string
        enum [on off]
      }
    }
    odmAction {
      on {}
      off {}
    }
  }
}
```

Thing Definition Language

ODM Terms in SDF

- SDF Keywords
- Object Qualities
- Action Qualities
- Event Qualities
- Property Qualities
- Data Qualities
  - ODM Data Qualities
  - JSON Schema Qualities
  - Structured Data Qualities (From JSON Schema)

SDF top level Keywords

- Info block
  - info, title, version, copyright, license
- Definitions block
  - namespace, defaultNamespace
  - odmObject, odmProperty, odmAction, odmEvent, odmData

Object qualities

```
"description": {
  "type": "string"
},
"title": {
  "type": "string"
},
"id": {
  "type": "string"
},
"name": {
  "type": "string"
},
"type": {
  "type": "string"
},
```

```
"optional": {
  "type": boolean"
{
"extends": {
  "type": "string"
},
"refines": {
  "type": "string"
},
"includes": {
  "type": "string"
},
```

Action Qualities

Event Qualities

Property Qualities

- Object Qualities
- All of the Data Qualities

```
"observable": {
  "type": "boolean"
},
"contentFormat": {
  "type": "string"
},
"readOnly": {
  "type": "boolean"
},
"writeOnly": {
  "type": "boolean"
}
```

Data Qualities

- Object Qualities
- All of the Data Qualities

```
"units": {
  "type": "string"
},
"scaleMinimum": {
  "type": "number"
},
"scaleMaximum": {
  "type": "number"
},
```

```
"nullable": {
  "type": "boolean"
},
"encoding": {
  "type": "object",
  "properties": {
    "widthInBits": {
    "type": "number"
    }
  }
}
```

Data Qualities – JSON Schema

```
"type": {
  "type": "string",
  "enum": [ "number", "string",
"boolean", "integer", "array",
"object" ]
  },
"enum": {
  "type": "array"
},
"const": {
  "type": { "oneOf" : [
"number", "string", "boolean",
"array", "object", "null" ] }
},
"default": {
  "type": { "oneOf" : [
"number", "string", "boolean",
"array", "object", "null" ] }
},
```

```
"pattern": {
  "type": "string"
},
"minimum": {
  "type": "number"
},
"maximum": {
"type": "number"
},
"multipleOf": {
"type": "number"
},
"maxLength": {
"type": "number"
},
"minLength": {
"type": "number"
},
```

Structured Data Qualities

```
"oneOf": {
  "type": "array",
  "minItems": 1
},
"anyOf": {
  "type": "array",
  "minItems": 1
},
"allOf": {
  "type": "array",
  "minItems": 1
},
"items": {
  "oneOf": [
    { "type": "array" },
    { "type": "object" }
  ]
},
```

```
"contains": {
  "oneOf": [
    { "type": "array" },
    { "type": "object" }
  ]
},
"minItems": {
  "type": "number"
},
"maxItems": {
  "type": "number"
},
"properties": {
  "type": "object"
},
```