

A Simple JSON Format for ODM Definitions

Michael Koster

April 17, 2019

Overview

- Simple definition format for the ODM ontology
 - Object
 - Event, Action, Property
 - Data types
- Defined namespaces using curie notation
- Flat definition space with cross-references
- One file per Object definition
- Examples of definitions

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://smarththings.example.com/capability/odm"
  },
  "defaultNamespace": "st",
  "object": {
    "Switch": {}
  },
  "property": {
    "Switch.value": {
      "type": "string",
      "enum": ["on", "off"]
    }
  },
  "action": {
    "Switch.on": {},
    "Switch.off": {}
  }
}
```

Simple example – Header Part

keywords

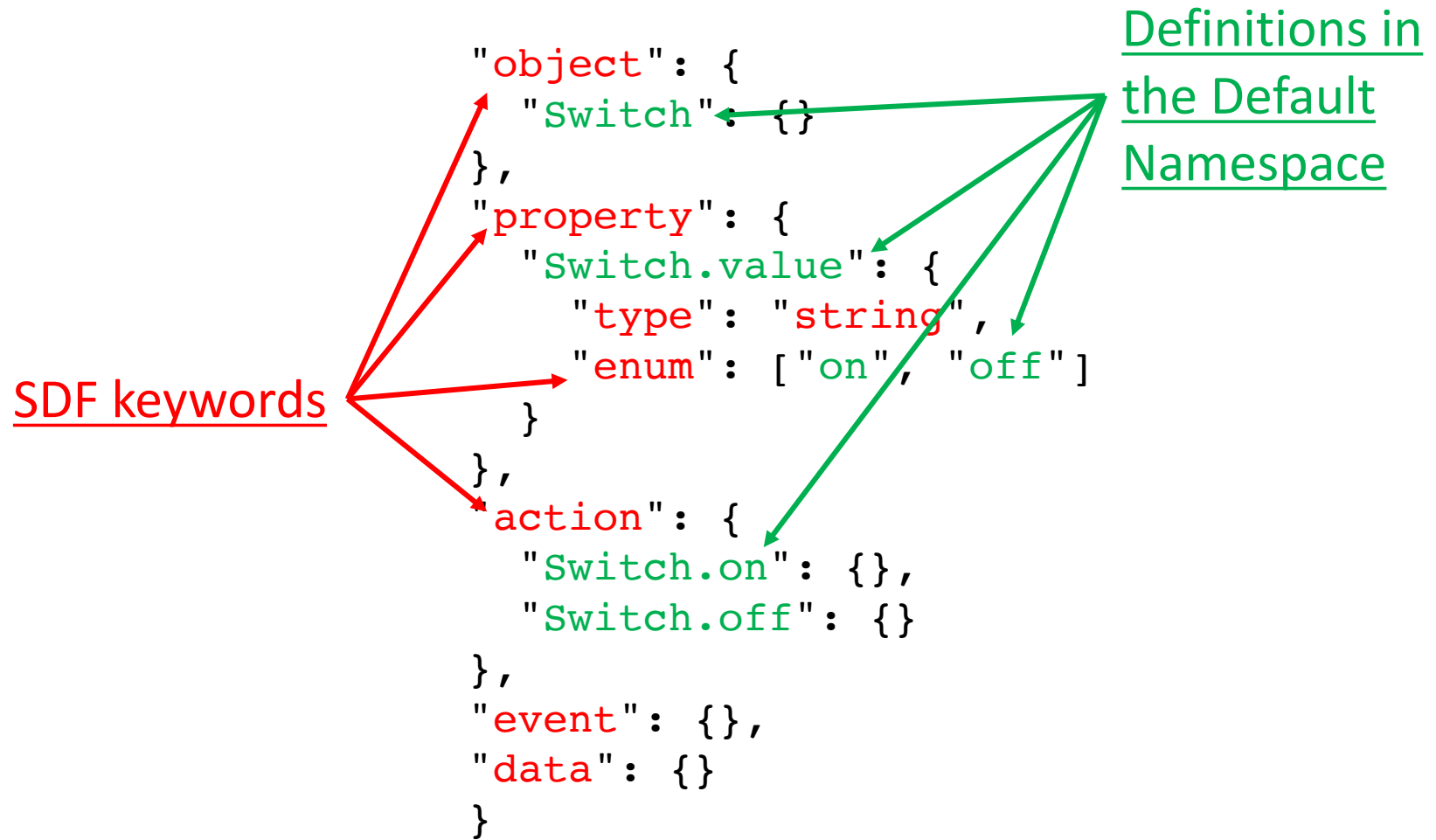
File Information

```
"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://openconnectivity.example.org/vocab/odm",  
  "st": "http://smarththings.example.com/capability/odm"  
},  
"defaultNamespace": "st",
```

Definitions



Definitions

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

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

Identifier namespace resolution

1 - Explicit namespace prefix (e.g. "st:Switch")

2 - SDF Keywords:

<https://github.com/mjkoster/ODM-Examples/blob/master/SDF2-Schema.json>

3 - defaultNamespace

SDF top level Keywords

- info
 - title, version, copyright, license
- namespace, defaultNamespace
- object, property, action, event, data (definitions)

Object qualities + common 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"  
},
```

Property Qualities

- All of the Object Qualities
- All of the Data Qualities

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

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

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"  
},
```

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"
},
"readOnly": {
  "type": "boolean"
},
"writeOnly": {
  "type": "boolean"
}
```

Action Qualities

- All of the Object qualities
- ?

Event Qualities

- All of the Object qualities
- ?

SDF Example

<https://github.com/mjkoster/ODM-Examples/SDF2.json>

```
{
  "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://smarththings.example.com/capability/odm"
  },
  "defaultNamespace": "st",
  "object": {
    "Switch": {}
  },
  "property": {
    "Switch.value": {
      "type": "string",
      "enum": ["on", "off"]
    }
  },
  "action": {
    "Switch.on": {},
    "Switch.off": {}
  }
}
```

SDF Example Definition

<https://github.com/mjkoster/ODM-Examples/blob/master/SDF2-SwitchLevel.json>

```
"object": {
  "SwitchLevel": {}
},
"property": {
  "SwitchLevel.level": {
    "type": "SwitchLevel.levelData",
    "readOnly": false
  }
},
"action": {
  "SwitchLevel.setLevel": {
    "data": [
      "SwitchLevel.levelData",
      "SwitchLevel.rateData"
    ]
  }
},
"data": {
  "SwitchLevel.levelData": {
    "type": "number",
    "minimum": 0,
    "maximum": 100,
    "multipleOf": 1
  },
  "SwitchLevel.rateData": {
    "type": "number",
    "minimum": 0,
    "maximum": 65535,
    "multipleOf": 1
  }
}
```


ZCL Example

```
"object": {
  "OnOff": {
    "id": "0x0006",
  }
"property": {
  "OnOff.OnOff": {
    "type": "zcl:OnOff.OnOffData",
    "readOnly": true,
    "optional": false
  }
}
"action": {
  { "OnOff.On": { "optional": false } },
  { "OnOff.Off": { "optional": false } },
  { "OnOff.Toggle": { "optional": true } }
},
"data": {
  "OnOff.OnOffData": {
    "type": "boolean",
    "default": false
  }
}
}
```

OCF Example

- ODM Object mapped to OCF Resource Type
- ODM Property mapped to OCF Property
- Action definitions composed from Property state
- Actions supply values for Properties e.g. on=true, off=false

OCF Example

```
"object": {  
  "BinarySwitch": {  
    "id": "oic.rt.BinarySwitch",  
    "description" : "This is a simple on/off switch."  
  }  
},  
"property": {  
  "BinarySwitch.value": {  
    "description": "The state of the switch.",  
    "type": "boolean",  
    "readOnly": false,  
    "writeOnly": false,  
    "optional": "false"  
  }  
}
```

OCF Example – Composed Actions

```
"property": {
  "BinarySwitch.value": {
    "type": "BinarySwitch.valueData"
  }
}
"action": {
  "BinarySwitch.on": {
    "data": {
      "type": "BinarySwitch.valueData",
      "const": true
    },
    "BinarySwitch.off": {
      "data": {
        "type": "BinarySwitch.valueData",
        "const": false
      }
    }
  }
}
"data": {
  "BinarySwitch.valueData",
  "type": "boolean"
}
```