IPSO Smart Objects

Jaime Jimenez*, Hannes Tschofenig†
*Ericsson, Email: Jaime.Jimenez@ericsson.com
†ARM Limited, Email: Hannes.Tschofenig@arm.com

I. Introduction

Standards for constrained devices are rapidly consolidating and the availability of IP on constrained devices enabled these devices to easily connect to the Internet. The IETF has also created a set of specifications for such IP-enabled devices to work in a Web-like fashion. One of such protocol is the Constrained Application Protocol (CoAP) [1] that provides request/response methods, ways to identify resources, discovery mechanisms, etc. similar to HTTP [2] but for use in constrained environments.

On top of protocols like CoAP and HTTP there is a need for structuring data as well.

IPSO Smart Objects provide a common design pattern, an object model, to provide high level interoperability between Smart Object devices and connected software applications on other devices and services. IPSO Objects are defined in such a way that they do not depend on the use of CoAP, any RESTful protocol is sufficient. Nevertheless, to develop a complete and interoperable solution the Object model is based on the Open Mobile Alliance Lightweight Specification (OMA LWM2M) [3], which is a set of management interfaces built on top of CoAP in order to enable device management operations (bootstrapping, firmware updates, error reporting, etc.). While LWM2M uses objects with fixed mandatory resources, IPSO Smart Objects use a more reusable design.

II. Data Model

The data model for IPSO Smart Objects consists of five parts:

1) Class Hierarchy
2) Schema
3) Data types
4) Operations
5) Content formats

A. Class Hierarchy

The URI template follows standard Web Linking [4] and standard CoRE Link [5] formats. It defines an implicit class hierarchy consisting of Objects and Resources. The Object class is further split into ID and Instance subclasses.

Objects are typed containers, which define the semantic type of instances. Instances inherit the type of their parent object, and allow Smart Object endpoints to expose multiple sensors and actuators of a particular type. Object instances are themselves containers for resources, which are the observable properties of an object.

This class hierarchy allows application software to use simple APIs. For complex objects, linking of an object to another object through an object link resource is allowed. This enables the recursion to be handled at the object level, using design patterns similar to web linking. An application client can consume a device's API without knowing it's structure and attributes a priori.

IPSO Smart Objects define the behaviour at the object and resource level, for example the tracking of minimum and maximum values, resetting of accumulated values, etc.

B. Schema

The URI Template defined in LWM2M is used for IPSO Smart Objects consists of three unsigned 16-bit integers separated by the character '/' in the following form *Object ID/Instance ID/Resource ID*

Semantically, the object type represents a single measurement, actuation, or control point, for example a temperature sensor, a light (actuator), or an on-off switch (control point).

The semantic meaning of a resource specifies a particular view or active property of an object. For example, a temperature sensor object might expose the current value (most recent reading), also the minimum and maximum possible reading, the minimum and maximum reading in an interval, and attributes like engineering units and application type.

Figure 1 shows an example URI of a temperature sensor.

Position paper for the 'IoT Semantic Interoperability Workshop 2016', 17^{th} and 18^{th} March 2016, San Jose, US. The content of this document describes the current state of work at the IPSO Alliance. The document has been reviewed and approved by the IPSO Smart Object Committee.

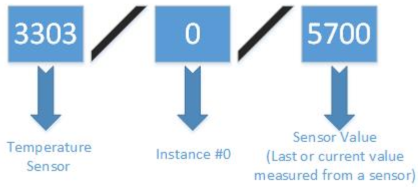

Fig. 1. Temperature Sensor URI Example.

Attributes describe the metadata configuration, settings, and state of an object or resource, and are discoverable by reading the link-format data of an object or resource. Multiple attributes may be serialized in the link-format descriptors that an object exposes.

Some attributes are immutable for a given object or resource type. For example, the static read, write, and execute capability attributes are derived from a Smart Object's definition file, while other attributes, like the LWM2M Notification Attributes, are used to dynamically configure a particular object instance or resource.

Attributes are represented using the IETF CoRE Link-Format (RFC 6690) or an equivalent mapping to other content formats, for example, application/json+ld.

C. Data Types

IPSO Smart Objects re-use the data types defined in the OMA LWM2M specification [3].

1) String: A UTF-8 string
2) Integer: An 8, 16, 32 or 64-bit signed integer.
3) Float: A 32 or 64-bit floating point value.
4) Boolean
5) Opaque: A sequence of binary octets.
6) Time: Unix Time.
7) Object Link: The object link is used to refer an instance of a given object.

D. Operations

IPSO Objects and their resources have the same operations as their counterparts in the OMA LWM2M specification [3] with the same semantics.

1) Resource values: Read, Write, Execute (restricted by the Access Type field)
2) Object Instances: Create, Delete (restricted by the Multiple Instances field)
3) Objects and their instances: Read, Write
4) Attributes: Set, Discover

E. Content Formats

Content formats are those specified by the OMA LWM2M specification [3]:

1) Resource values: text/plain, tlv
2) Objects: text/senml+json, application/cbor, binary/tlv
3) Attributes: link-format, link-format+json

III. Composite Objects

As devices increase in complexity (e.g., from a sensor to an appliance, from a switch to a complex actuator) the need to link resources to create more complex objects or "Composite Objects" arises. Such a composite object can, for example, be constructed with a single reusable type "generic composite object" with one ID. The resources may be of a generic reusable link type, also using a single ID, with multiple instances allowed. For example, '4000/0/6700/0' where 4000 is a "composite object" and 6700 is "generic object link". Composite objects offer higher granularity than one large nested object would. An observer of a device represented as a composite object could reduce bandwidth utilization by observing only the linked object instances instead of the full object. Figure 2 shows an example.

IV. Humidity Sensor Example

Specification authors use different ways to describe resources exposed by IoT devices. Often natural text is used and sometimes more formal techniques are relied on. For the definition of the IPSO Smart Objects tables with natural language descriptions and XML (to offer machine-readable descriptions) are used.

The following is an example of a humidity sensor that contains the sensor value, units, min and max measured values, min and max range values and a resets for those. Figure 3 shows the object and resource definitions in a tabular form and the definition in XML is shown in Appendix A.

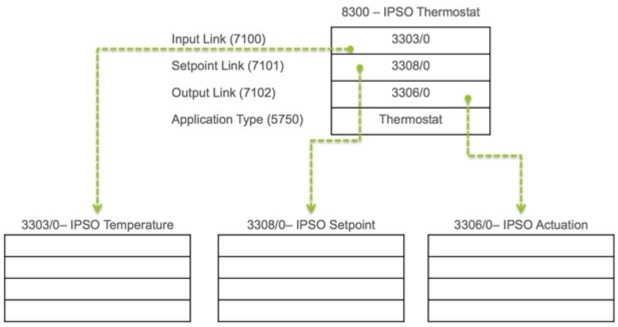

Fig. 2. Composite Object Example.

Object	Object ID	Object URN	Multiple Instances?	Description
IPSO Humidity	3304	**urn:oma:lwm2m:ext:3304**	Yes	Relative humidity sensor, example units = %

Resource Name	Resource ID	Access Type	Multiple Instances?	Mandatory	Type	Range or Enumeration	Units	Descriptions
Sensor Value	5700	R	No	Mandatory	Float			Last or Current Measured Value from the Sensor
Units	5701	R	No	Optional	String			Measurement Units Definition e.g. "Cel" for Temperature in Celsius.
Min Measured Value	5601	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The minimum value measured by the sensor since power ON or reset
Max Measured Value	5602	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The maximum value measured by the sensor since power ON or reset
Min Range Value	5603	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The minimum value that can be measured by the sensor
Max Range Value	5604	R	No	Optional	Float	Same as Measured Value	Same as Measured Value	The maximum value that can be measured by the sensor
Reset Min and Max Measured Values	5605	E	No	Optional	Opaque			Reset the Min and Max Measured Values to Current Value

Fig. 3. IPSO Humidity Object Definition.

V. IPSO Smart Objects

A. Starter Pack

This IPSO Smart Objects Starter Pack describes 18 Smart Objects, including a temperature sensor, a light controller, an accelerometer, a presence sensor, and other types of sensors and actuators. These objects are common in many application domains. Appendix B shows the list of objects defined in the Starter Pack.

With this initial publication the IPSO Alliance aimed to offer developers and standardization experts a starting point from which to build more complex objects in order to address vertical IoT market segments. One important design criteria in the design of the IPSO Smart Objects was and is to making it easy for developers to derive new objects based on their use case needs, while promoting interoperability to an extent as is practical. Naturally, a device that has not seen a newly defined object cannot know the semantics even if the contained resources can be understood on a syntactical level. However, re-use of existing object and resource definitions allows application developers to re-use code

B. Expansion Pack

To complement the initial set of objects, the IPSO Smart Object Expansion Pack was published. The Expansion Pack adds 16 common template sensors, 6 special template sensors, 5 actuators and 6 control switch types.

Some of the newly defined objects are generic in nature, such as voltage, altitude or percentage, while others are more specialized like the Color Object or the Gyrometer Object. New actuators and controllers are defined such as timer, buzzer, joystick and level. All of these objects were found to be necessary on a variety of use case domains.

Appendix C lists the objects defined in the IPSO Expansion Pack.

C. Extensibility

Apart from the objects published by the IPSO Alliance developers and standardization experts are encouraged to define new objects tailored to their use cases if the already available functionality is insufficient.

The Starter and Expansion Pack provide basic examples for common sensors and actuators. Developers may extend the existing object set and submit them to IPSO.

References

[1] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," Internet Engineering Task Force, RFC 7252, Jun. 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7252.txt

[2] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," Internet Engineering Task Force, RFC 2616, Jun. 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2616.txt

[3] O. M. Alliance, "Lightweight Machine-to-Machine Technical Specification v1.0, Candidate Enabler," Dec. 2015. [Online]. Available: http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/oma-lightweightm2m-v1-0

[4] M. Nottingham, "Web Linking," Internet Engineering Task Force, RFC 5988, Oct. 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5988.txt

[5] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format," Internet Engineering Task Force, RFC 6690, Aug. 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6690.txt

Appendix

Appendix A: Humidity Object Definition in XML Format

The following is the definition document for the Humidity Object in XML.


```
<?xml version="1.0" encoding="utf-8"?>
<LWM2M>
    <Object ObjectType="MODefinition">
        <Name>Humidity</Name>
        <Description1>This IPSO object should
        be used with a humidity sensor to report a humidity
        measurement.  It also provides resources for
        minimum/maximum measured values and the minimum/maximum
        range that can be measured by the humidity sensor.
        An example measurement unit is relative humidity as a
        percentage (ucum:%).
        </Description1>
        <ObjectID>3304</ObjectID>
        <ObjectURN>urn:oma:lwm2m:ext:3304</ObjectURN>
        <MultipleInstances>Multiple</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Resources>
            <Item ID="5700">
                <Name>Sensor Value</Name>
                <Operations>R</Operations>
                <MultipleInstances>Single</MultipleInstances>
                <Mandatory>Mandatory</Mandatory>
                <Type>Float</Type>
                <RangeEnumeration></RangeEnumeration>
                <Units>Defined by "Units" resource.</Units>
                <Description>Last or Current Measured Value from
                the Sensor
                </Description>
            </Item>
            <Item ID="5601">
                <Name>Min Measured Value</Name>
                <Operations>R</Operations>
                <MultipleInstances>Single</MultipleInstances>
                <Mandatory>Optional</Mandatory>
                <Type>Float</Type>
                <RangeEnumeration></RangeEnumeration>
                <Units>Defined by "Units" resource.</Units>
                <Description>The minimum value measured by the
                sensor since power ON or reset
                </Description>
            </Item>
            <Item ID="5602">
                <Name>Max Measured Value</Name>
                <Operations>R</Operations>
                <MultipleInstances>Single</MultipleInstances>
```

```
        <Mandatory>Optional</Mandatory>
        <Type>Float</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units>Defined by "Units" resource.</Units>
        <Description>The maximum value measured by the
        sensor since power ON or reset
        </Description>
    </Item>
    <Item ID="5603">
        <Name>Min Range Value</Name>
        <Operations>R</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>String</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units>Defined by "Units" resource.</Units>
        <Description>The minimum value that can be measured
        by the sensor
        </Description>
    </Item>
    <Item ID="5604">
        <Name>Max Range Value</Name>
        <Operations>R</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>Float</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units>Defined by "Units" resource.</Units>
        <Description>The maximum value that can be measured
        by the sensor
        </Description>
    </Item>
    <Item ID="5701">
        <Name>Sensor Units</Name>
        <Operations>R</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>String</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>Measurement Units Definition e.g. "Cel"
        for Temperature in Celsius.
        </Description>
    </Item>
    <Item ID="5605">
        <Name>Reset Min and Max Measured Values</Name>
        <Operations>E</Operations>
        <MultipleInstances>Single</MultipleInstances>
        <Mandatory>Optional</Mandatory>
        <Type>Opaque</Type>
        <RangeEnumeration></RangeEnumeration>
        <Units></Units>
        <Description>Reset the Min and Max Measured Values to
        Current Value
        </Description>
    </Item>
</Resources>
<Description2></Description2>
```

```
    </Object>
</LWM2M>
```

Appendix B: IPSO Starter Pack

The IPSO Starter Pack defines the objects shown in Table I.

TABLE I
IPSO Starter Pack.

Object	Object ID
Digital	3200
Digital Output	3201
Analogue Input	3202
Analogue Output	3203
Generic Sensor	3300
Illuminance Sensor	3301
Presence Sensor	3302
Temperature Sensor	3303
Humidity Sensor	3304
Power Measurement	3305
Actuation	3306
Set Point	3308
Load Control	3310
Light Control	3311
Power Control	3312
Accelerometer	3313
Magnetometer	3314
Barometer	3315

Appendix C: IPSO Expansion Pack

The IPSO Expansion Pack defines the objects shown in Table II.

TABLE II
IPSO Expansion Pack.

Object	Object ID
Voltage	3316
Current	3317
Frequency	3318
Depth	3319
Percentage	3320
Altitude	3321
Load	3322
Pressure	3323
Loudness	3324
Concentration	3325
Acidity	3326
Conductivity	3327
Power	3328
Power Factor	3329
Rate	3346
Distance	3330
Energy	3331
Direction	3332
Time	3333
Gyrometer	3334
Color	3335
GPS Location	3336
Positioner	3337
Buzzer	3338
Audio Clip	3339
Timer	3340
Addressable Text Display	3341
On/Off Switch	3342
Push Button	3347
Level Control	3343
Up/Down Control	3344
Multistate Selector	3348
Multiple Axis Joystick	3345