

Position Paper for IoT Semantic Interoperability Workshop 2016

YANG-Based Constrained Management Interface (CoMI)

Peter van der Stok consultancy@vanderstok.org

Andy Bierman <andy@yumaworks.com>

2016-02-21

IoT devices do not exist as stand-alone processors, but are associated with equipment. A resource model, hosted by the IoT device, enables the manipulation of the connected equipment. This position paper argues that the use of the YANG language can be extended for use in IoT network management and device management. YANG can be used to describe the resource interface hosted by the IoT device.

1) YANG data modelling language for IoT

The YANG data modeling language [RFC6020] is gaining popularity in the IETF and the networking industry because it is easy to learn and capable of describing complex programmatic interfaces. Although originally designed for use with NETCONF [RFC 6241], it is also being applied to HTTP/REST in new protocols, such as RESTCONF [draft-ietf-netconf-restconf], which maps YANG defined data to the REST resource model and HTTP CRUD operations.

YANG is designed to provide an “off-line” schema definition shared by a client and server. Consequently, protocols that use YANG do not waste bandwidth coming from servers sending lots of meta-data to each client, describing the data models it supports. A YANG data model module is an “API contract” between client and server. It needs to be defined in advance, not at run-time. It is important that this contract be as detailed and accurate as possible. The YANG data module is the description of the API contract, not the implementation itself. The modeling and automation tools might be complex, but they are only needed during development. The run-time on-device complexity is not coupled to model complexity.

YANG provides many different mechanisms to address IoT devices:

- provides an easy to learn and easy to read syntax
- supports multiple message transport formats (XML, JSON, CBOR)
- provides rich combination of human and machine-readable statements
- provides hierarchical and reusable data structures that can match internal structures
- supports distributed authorship, so modules from different sources can be easily used together
- supports language extensions and model extensions by anybody, without centralized control

- provides lifecycle, conformance model, platform-specific model deviations for real deployment

Embedded IoT devices have interfaces to manipulate the connected equipment. All relevant data model interfaces need to be known to the software engineers during development, in order for the client business logic to properly manage the resources on each device. YANG is rich enough to express the properties of the data model.

YANG serves as both an information model and a data model in actual deployment, because it is designed to support language extensions. These “external statements” are used by tool vendors, standard YANG modules, protocol definitions, etc. to adapt YANG beyond its original stated purpose for NETCONF. These statements help drive tool automation, which is a critical component of the success of YANG. This allows vendors to minimize the cost of device management software development and also maintain consistent, stable behavior using centralized, reusable software components. This also helps reduce code size when the number of YANG modules implemented on a device grows over time.

2) Constrained IOT devices

A powerful and extensible data model is needed as the basis of IoT device management, in order to help developers create device instrumentation and operators manage constrained networks. The CORE WG in the IETF is planning to adapt RESTCONF and YANG for use with the Constrained Application Protocol (CoAP) [RFC 7252]. CoAP is designed to be extremely efficient in constrained networks and it can be efficiently implemented on constrained nodes.

The CORE WG is considering the needs of many different use-cases for the network management protocol for CoAP. The Constrained Management Interface (CoMI) [draft-vanderstok-core-comi] is designed to retain the benefits of RESTCONF and YANG, but in a protocol optimized for constrained IoT devices:

- Replace use of HTTP with CoAP.
- Add PATCH method to CoAP [draft-vanderstok-core-patch].
- Replace use of TLS/TCP with DTLS/UDP.
- Adapt RESTCONF Use of HTTP Methods to CoAP.
- Adapt YANG RPCs and notification events to CoAP.
- Replace XML and JSON with CBOR binary encoding [RFC 7049].
- Define direct YANG to CBOR encoding rules.
- Replace XPath object identifiers with optimized numeric identifiers:
 - YANG Hash: calculated 30-bit IDs [draft-bierman-core-yang-hash],

- Managed IDs: assigned identifiers (1 – 4 byte IDs) [draft-veillette-core-cool].
- YANG Module Discovery through sharable YANG library [draft-ietf-netconf-yang-library].
- Optional Multi-Resource Editing with YANG Patch [draft-ietf-netconf-yang-patch].

3) Device management

Standardization organizations define interfaces hosted by processors to manipulate the connected equipment. Examples of such standardization organizations are BACnet, KNX, ZigBee, oBIX, OMA, and many others. These organizations plan to move to resource based interfaces. The data models proposed by these organizations can be accurately expressed by YANG. Since YANG allows any module to extend any other module, it is well-suited for real-world deployments in which standard and vendor data are inter-twined and important for product differentiation and experimentation. It is possible that some data models intended for non-constrained devices will be needed on some IoT devices. These modules should be usable without modification. YANG can be the common language for defining all device instrumentation. The protocols will be different depending on use-cases, but the information embodied in the YANG modules remains uniform and consistent across all protocols.

4) Conclusion

In summary, the YANG data modeling language can be used to define a resource interface for device management. The powerful extensibility mechanisms in YANG help drive protocol and data model automation tools to reduce development costs and improve consistency. The RESTCONF protocol provides an automatic mapping between YANG data and HTTP/REST methods. The CoMI protocol captures the features of YANG and RESTCONF, and targets YANG to constrained IoT devices. The result combines powerful data model driven automation with the device and network efficiency needed in constrained IoT devices.