NETMOD
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

D. Bogdanovic
Juniper Networks
B. Claise
C. Moberg
Cisco Systmes, Inc.

March 09, 2015

YANG model classification
draft-bogdanovic-netmod-yang-model-classification-01

Abstract

YANG became de facto standard language for data modeling in the industry. More and more groups uses YANG to create protocol and service models, both for configuration and operational models. Currently there is a lack of consistent terminology to categorize those models. A consistent terminology would help models categorization, assist in the analysis the YANG data modeling effort in the IETF and in the industry, and facilitate the YANG-related discussions between different groups.

Table of Contents

1. Introduction

With more and more data models being described with YANG [RFC6020], a taxonomy for the models is needed. In this draft, authors try to propose a taxonomy for YANG models.

2. Problem Statement

YANG [RFC6020] became de-facto standard language for data modeling in the industry. Not only at the IETF, but also in multiple Standard Development Organizations, different consortia, ad hoc groups, and OSP. Therefore, many YANG models are being developed and published. Today, there is no classification of models, there are no clear guidelines on how to layer models on each other, or how to classify existing or new models. With this document, the authors are proposing a new way for YANG model classifications.

Acknowledging that the YANG became the de-facto standard language for data modeling, the Internet Engineering Steering Group (IESG) has been encouraging the working groups to use the NETCONF [RFC6241] and YANG standards for configuration, especially in new charters [Writable-MIB-Module-IESG-Statement].

YANG Models can classified according to two dimensions: based on the layer in the hierarchy of models, and based on the model type. Those two categories are covered in the next two sections.

3. First Dimension: Network YANG Data Model Layering

When developing models, there are two approaches possible, top down and bottom up. Top down approach is driven by business requirements and bottom up is driven by technological ones.

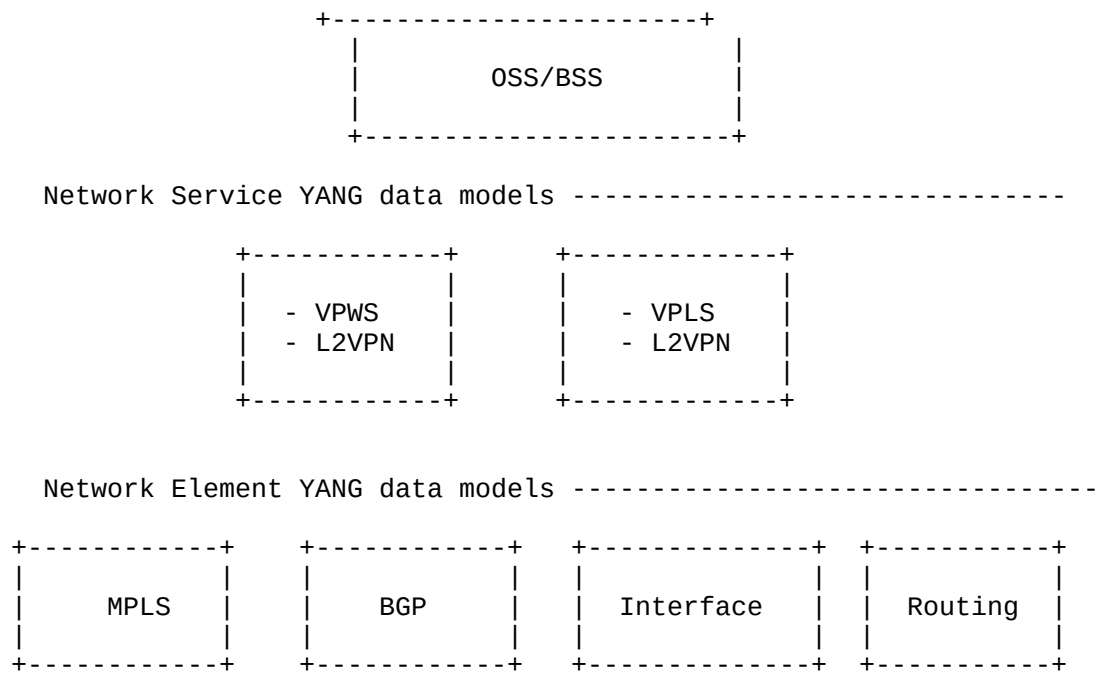
There are no hard requirements on how to create modeling, but it would be useful to have a classification and to create models that can be easily reused, as with this time and energy will be saved in future development. We should stimulate both development styles, bottom up and top down, as each has its benefits and groups to which a certain style will be more appealing than the other.

For layering purposes, we can classify data models into two layers:

Network Service YANG Model

Network Element YANG Model

Figure 1 displays example YANG models at different layers. By layering the models, it is easier to achieve reusability of existing lower layer models in higher level models and preventing duplication of same features modeled in different layers. When developing models per layers, it allows creating very focused groups in specific areas. As an example, creating protocol data definitions network equipment YANG data models should involve people that have intimate experience of implemention details. On the other hand, network service models are best developed by people experienced in network operations. Same network service, can be implemented and modeled using different protocol and feature YANG models.



3.1. Network Service YANG Data Models

Network Service YANG models are created by network operators when they choose how to configure their network from technology point of view. They decide which technology is best match for their business needs and based on that create network service data models. With more and more configuration models being available, both vendor and standards one, network service model developer can create reusable components based on the configuration models, and use those service components data models to create end to end service model.

For example, http://datatracker.ietf.org/doc/draft-l3vpn-service-yang/ provides an abstracted view of the Layer 3 IP VPN service configuration components. It will be up to an orchestrator to take this as an input and use specific configurations models on the network elementlayer to configure the different network elements to deliver the service.

Network Service YANG models can be developed in multiple ways. Building them monolithic from vendor models or by combining one or more service components into an end to end service data model. It specifies complete service that is provided by the network operator.

Building monolithic network service model has an advantage of doing it fast, but at the expense of flexibility of updating the service later or equipment vendors. Such an end to end service can be VPLS/ VPWS L2VPN, IPsec, etc. If we take into example VPLS L2VPN service, it can be built as a single network service model or it can be built from several service components. VPWS L2VPN service can be built on top of MPLS or IP network core. When building such a network service model, network variations have to be taken into the account and by creating service components modle, such as MPLS, BGP service component models, it is easier to build a network service model, such as VPWS L2VPN.

3.2. Network Element YANG Data models

This is base model for all higher models. It fully describes protocol, such as OSPF [I-D.yeung-netmod-ospf], ISIS [I-D.ietf-isis-yang-isis-cfg] or feature, example access control list [I-D.ietf-netmod-acl-model]. The base model can be either vendor specific, which then describes vendor implementation of the protocol or feature or standard model. Key difference between those two is what is implemented. Vendor device model will always describe what is implemented, which can ber more or even less then in standard model. Standard model describes what is agreed in the industry to be accepted as base description.

4. Second Dimension: Model Type

At very high level, models can be divided into proprietary and standard. Each vendor, consortium, open source project can publish their models and those are considered proprietary models. When an SDO, such as IETF or ITU, publishes an accepted model document, then this is a standard model. There are use cases where a consortium has published work which de facto became standard, such as Linux kernel, but for the clarity in this document, authors are making a separation between models based on the above description.

Standard YANG Model:

Standard Extension YANG Model: .

Proprietary Extension to Standard YANG Model: As the Standard YANG Models contains a subset of all the Vendor Configuration Models, proprietary extensions must complement the Standard YANG Models to represent a Vendor Configuration Model.

Vendor Configuration Model: It describes all configurable capabilities of the device and what device vendor exposes for

configuration. The vendor configuration model can be CLI or YANG-based.

Proprietary YANG Model: A non Standard YANG Model.

As mentioned earlier in this document, there are two ways of designing models, top down and bottom up with one restriction. Everything is dependent on the vendor data model. That model describes all the possibilities and if model developers prefers, they can use vendor model only to design service components, network service and business service. Using vendor model provides all capabilities today, but it comes with restrictions of portability between vendors and to certain extent devices. On the other hand, only standard models and standard extensions can be used, but this might result in less feature rich or less efficient services. Service model developer has a choice to reuse service components or write a model completely based on vendor data model.

4.1. Standard YANG model

With YANG we have a common language, that enables different communities to express data models that are widely understandable without lot of additional explanation. This enables different groups, such as IETF, to standardize data models, defined as an IETF RFC, and vendors to support them, which will make it easier to for network operators to manage their network configuration programmatically. For example, A YANG Data Model for Interface Management [RFC7223], or the Configuration Data Model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols [RFC6728].

4.2. Standard Extension YANG Model

Standard Extension is a the conditional portion of a Standard YANG Model, expressed with the feature, if-feature, augment YANG statements [RFC6020]. An example of such standard extension is policy based routing (PBR). PBR is found in many vendor implementations and have many common features, but not all vendors support PBR on all of their devices.

4.3. Proprietary Extension to Standard YANG Model

Proprietary extension is a conditional portion of a Standard YANG Model, expressed with feature, if-feature, augment YANG statements [RFC6020]. Proprietary extension can be a feature depending on harward platform capabilities and it is not available by other vendors. Such an example could be match condition for packet classification used for PBR.

4.4. Vendor configuration model

Base model for all other models is the vendor configuration model. It describes all configurable capabalities of the device and what device vendor exposes for configuration.

The standard configuration model is a subset of vendor configuration model. The standard configuration model can be broken into base model and standard extension models, where the base is common data model and standard extensions are standard features that are not implemented by all vendors. Example of standard base model is Access Control List and routing filter is a standard extension on ACL. Or another example: encryption algorithm is standard feature, but the different types, like md5, hmac-md5, hmac-sha1, etc are standard extensions, as it is not that all vendors have all encryption algorithm types implemented.

Although all vendors provide very similar functionality using standards, implementations are different. One of basic examples are dynamic routing protocols. We can see today two main types of routing protocol configuration.

protocol centric - all the protocol related config is contained with the protocol itself. Especially in case of multiple instances of the routing protocol running in different routing-instances (routing-instance as described in core routing model [I-D.ietf-netmod-routing-cfg]), all the routing-instance protocol config is contained in the default routing instance.

```
Router ospf 10
Default-metric 100
Address-family ipv4 vrf VRF1
    Network x.x.x.x area 0

Address-family ipv4 vrf VRF2
    Network x.x.x.x area 0

Address-family ipv4
    Network x.x.x.x area 1
```

In term of YANG model, the routing protocol configuration will be defined within the default routing-instance and the routing-protocol config will contain multiple instances referring to other routing-instances.

VRF centric - All the protocol related config for a routing-instance is contained within this routing-instance.

```
Routing-instance VRF1 {
  Protocols isis {

  }
}
Routing-instance VRF2 {
  Protocols isis {

  }
}
```

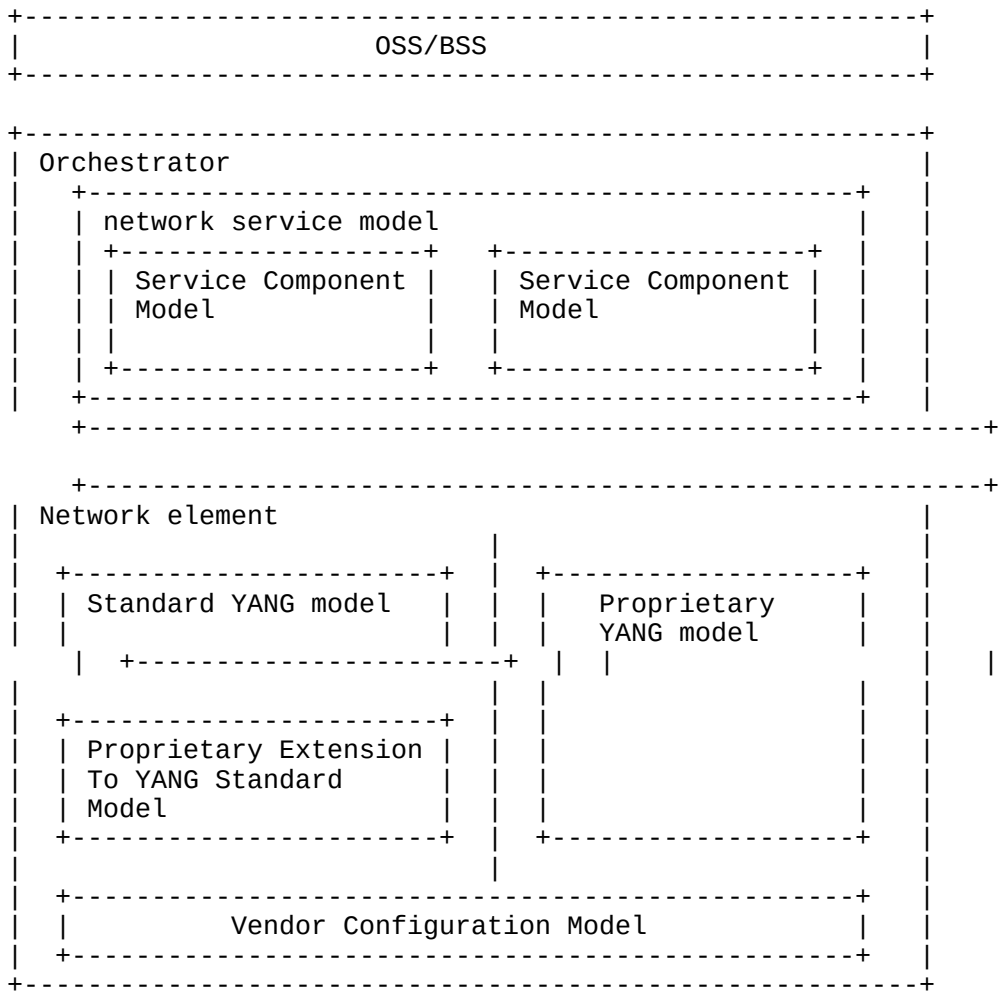
In term of YANG model, the routing protocol configuration for a routing-instance will be defined within the associated routing-instance.

The bottom line message is that, even if YANG models are standardized, they will provide different CLI outcomes, simply because the CLI among vendors is not standardized.

4.5. Proprietary YANG Model

While waiting for the Standard YANG Models to be published, the different vendors might offer Proprietary YANG Models.

5. Typical Architecture



6. IETF, Other SDOs, and open source

IETF, as a standard defining organization (SDO), is well positioned to standardize Network Element YANG models. With a wide range of expertise found within its WGs focused on those technology definitions. As IETF participants implement those protocols, they have deep expertise about the implementation and finding a common base standard configuration model between vendors should be a very viable goal.

In some situation where the protocols are standardized by different SDOs, those SDOs should be responsible for its YANG data modeling

effort. For example, the IETF has transferred the responsibility for some IEEE technology-related MIB modules to the IEEE 802.1 and 802.3 Working Group [RFC4663], [RFC7448]. Similarly, the IEEE should be responsible for similar YANG data modeling efforts.

Although many network operators participate in IETF work, developing higher level models requires network operations expertise. If such teams within right WG can be formed, then some service models can be developed within IETF, but some groups, like Metro Ethernet Forum or CableLabs could be better positioned for service modeling.

Today there are many open source projects and some of them are becoming de facto standards, like the Linux kernel. Many such open source projecs, like Open Daylight, OpenStack, etc, are doing very good work and their work is being accepted and deployed in production environments. They bring a lot of very valuable experience to other groups. From IETF perspective, if there is such a work present, it can be used as a very good starting point for modeling within IETF.

7. Security Considerations

At this stage, authors of the draft didn’t look into security considerations.

8. IANA Considerations

This document requests no action by IANA.

9. Acknowledgements

Thanks to David Ball for his enlightenments on Metro Ethernet Forum service aspects.

10. Change log [RFC Editor: Please remove]

version 1: restructure the document, add the two dimensions, add the interaction with the different SDOs and opensource projects, add the definitions.

11. References

11.1. Normative References

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

11.2. Informative References

[I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for ISIS protocol", draft-ietf-isis-yang-isis-cfg-01 (work in progress), October 2014.

[I-D.ietf-netmod-acl-model]
Bogdanovic, D., Sreenivasa, K., Huang, L., and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-01 (work in progress), February 2015.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-16 (work in progress), October 2014.

[I-D.yeung-netmod-ospf]
Yeung, D., Qu, Y., Zhang, J., Bogdanovic, D., and K. Sreenivasa, "Yang Data Model for OSPF Protocol", draft-yeung-netmod-ospf-02 (work in progress), October 2014.

[RFC4663] Harrington, D., "Transferring MIB Work from IETF Bridge MIB WG to IEEE 802.1 WG", RFC 4663, September 2006.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

[RFC6728] Muenz, G., Claise, B., and P. Aitken, "Configuration Data Model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols", RFC 6728, October 2012.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, May 2014.

[RFC7448] Taylor, T. and D. Romascanu, "MIB Transfer from the IETF to the IEEE 802.3 WG", RFC 7448, February 2015.

[Writable-MIB-Module-IESG-Statement]
"Writable MIB Module IESG Statement", <https://www.ietf
.org/iesg/statement/writable-mib-module.html>.

Authors' Addresses

Dean Bogdanovic
Juniper Networks

Email: deanb@juniper.net

Benoit Claise
Cisco Systmes, Inc.

Email: bclaise@cisco.com

Carl Moberg
Cisco Systmes, Inc.

Email: camoberg@cisco.com